

Symdyn: An automated algebraic solution for high-order quantum systems

D. Martínez-Tibaduiza^{1,*}, Vladimir Vargas-Calderón², J. G. Dueñas³, J. Flórez-Jiménez⁴, and A. Z. Khoury¹

¹*Instituto de Física, Universidade Federal Fluminense, Avenida Litorânea, 24210-346 Niterói, RJ, Brazil*

²*D-Wave Systems Inc., Burnaby, British Columbia, Canada*

³*Information Analysis Group, Special Jurisdiction for Peace, Cra 7 No 63-44, Bogotá, Colombia*

⁴*Lyric, Sunnyvale, California 94087, USA*



(Received 31 March 2025; accepted 15 December 2025; published 26 January 2026)

Many important quantum systems are characterized by Hamiltonians expressible as a linear combination of time-independent generators of a finite Lie algebra, $\hat{H}(t) = \sum_{l=1}^L \eta_l(t) \hat{g}_l$. The Wei-Norman framework provides a powerful analytical method for computing the exact time-evolution operator in a factorized form, $\hat{U}(t) = \prod_{l=1}^L e^{\Lambda_l(t) \hat{g}_l}$. By exploiting the underlying algebraic structure, the method yields a system of differential equations—the Wei-Norman equations—which determine the time-evolution operator coefficients $\Lambda_l(t)$ from the Hamiltonian ones $\eta_l(t)$. Although robust and formally extendable, applying the method by hand becomes increasingly difficult or even intractable for high-order systems ($L \geq 6$). In this work, we introduce *symdyn*, a Python library that automates the application of this method. Given the commutators between generators, the library symbolically derives the Wei-Norman equations and efficiently computes nested commutators, similarity transformations, and the differential equations that yield Baker-Campbell-Hausdorff-like relations. We validate the library by recovering known results for quantum systems associated with the Lie algebras $\mathfrak{su}(1, 1)$, $\mathfrak{su}(2)$, $\mathfrak{sl}(2)$, and $\mathfrak{so}(2, 1)$, and further demonstrate its capabilities by deriving the Wei-Norman equations for a system of two coupled time-dependent harmonic oscillators involving 11 generators. Additionally, we specialize the library to handle the Lie group $SU(N)$ using a generic Cartan-Weyl basis for its algebra, and demonstrate its versatility by deriving the Wei-Norman equations for $\mathfrak{su}(2)$, $\mathfrak{su}(3)$, and $\mathfrak{su}(4)$, including the explicit construction of a set of quantum gates for universal quantum computation.

DOI: [10.1103/24r3-j9zy](https://doi.org/10.1103/24r3-j9zy)

I. INTRODUCTION

The time evolution of quantum systems governed by time-dependent Hamiltonians is a fundamental problem in physics, with applications ranging from quantum optics and control theory to quantum thermodynamics and quantum computing. The Wei-Norman framework stands as a powerful algebraic method for obtaining exact solutions when the Hamiltonian can be expressed as a linear combination of linearly independent, time-independent generators $\{\hat{g}_l\}_{l=1}^L$ of a finite Lie algebra,

$$\hat{H}(t) = \sum_{l=1}^L \eta_l(t) \hat{g}_l, \quad (1)$$

where L is the algebraic order and the coefficients $\eta_l(t) \in \mathbb{C}$ are time-dependent functions. Any such linear combination defines a vector in the Lie algebra, which we refer to here as a *Lie vector*. Since the algebra is closed under commutation,

the commutator of any pair of generators yields a Lie vector

$$[\hat{g}_i, \hat{g}_j] = \sum_{l=1}^L \gamma_{ij}^l \hat{g}_l, \quad (2)$$

where the scalars γ_{ij}^l are the structure constants of the algebra.

The system's state vector $|\psi(t)\rangle$ evolves according to the Schrödinger equation, $i \frac{d}{dt} |\psi(t)\rangle = \hat{H}(t) |\psi(t)\rangle$ [1], with $\hbar = 1$ assumed. The time-evolution operator (TEO) relates the initial state vector $|\psi(t_o)\rangle$ to the state at any latter time via $|\psi(t)\rangle = \hat{U}(t, t_o) |\psi(t_o)\rangle$. It also fulfills the initial condition $\hat{U}(t \rightarrow t_o, t_o) = \hat{\mathbb{1}}$, where $\hat{\mathbb{1}}$ is the identity operator, and evolves according to the operator Schrödinger equation

$$i \frac{d}{dt} \hat{U}(t, t_o) = \hat{H}(t) \hat{U}(t, t_o). \quad (3)$$

Since $\hat{H}(t)$ lies in a Lie algebra, the corresponding TEO belongs to the associated Lie group and can be written in one of two main forms. The first, due to Magnus [2,3], is the unfactorized representation, in which the TEO is expressed as a single exponential

$$\hat{U}(t) = \exp \left(\sum_{l=1}^L \lambda_l(t) \hat{g}_l \right), \quad (4)$$

where $\lambda_l(t) \in \mathbb{C}$. To simplify the notation, we omit the initial time argument from this point forward. The second is the factorized representation, written as a product of exponentials

$$\hat{U}(t) = \prod_{l=1}^L e^{\Lambda_l(t) \hat{g}_l}, \quad (5)$$

*Contact author: danielmartinezt@gmail.com

Published by the American Physical Society under the terms of the [Creative Commons Attribution 4.0 International](https://creativecommons.org/licenses/by/4.0/) license. Further distribution of this work must maintain attribution to the author(s) and the published article's title, journal citation, and DOI.

where $\Lambda_l(t) \in \mathbb{C}$. The ordering of the exponentials in this product affects the functional form of the coefficients $\Lambda_l(t)$, though not the operator itself. Note that the generators do not need to be Hermitian, even if the Hamiltonian is required to be Hermitian to guarantee a unitary evolution. This factorized representation was developed by Wei and Norman in the 1960s [4,5] and has proven particularly useful for computing the quantum state vector and for interpreting the action of each exponential component of the TEO—associated with a given generator—on specific states.

The procedure to derive the coefficients Λ_l from the Hamiltonian coefficients η_l is known as the Wei-Norman method (WNM), yielding the Wei-Norman equations and the exact TEO. Conversely, extracting Λ_l from the coefficients λ_l of the unfactorized representation is known as Wei-Norman factorization, which leads to the so-called BCH-like relations (BCH from Baker-Campbell-Hausdorff). For a given algebra, both procedures result in structurally similar systems of nonlinear differential equations. Their solution depends on the chosen basis for expressing the algebra—that is, the specific set of generators—and their structure is influenced by the ordering of the exponentials in the factorization. Notably, the Wei-Norman framework is fully analytic, requiring no numerical or perturbative approximations.

While the factorized representation is globally valid for all solvable Lie algebras, there is no such guarantee for semisimple Lie algebras, where finding an optimal basis to simplify the problem is often nontrivial. Fortunately, for all classical simple Lie algebras, which are relevant to this work, a Cartan-Weyl basis provides a globally valid solution, in which the Wei-Norman equations take the form of a hierarchy of block-decoupled matrix Riccati equations [6,7]. Given the wide range of physical and mathematical systems described by such algebras, the WNM—also known as the decoupling or disentangling method—has emerged as a paradigmatic approach for obtaining exact group dynamics under arbitrary time-dependent modulation of the algebra vectors. Applications of this method span a wide range of fields, including optimal control [8–10], quantum dynamics [11–16], and, more recently, the study of open quantum systems [15]. A comprehensive tutorial covering the phase-space formulation and applications in quantum optics can be found in Ref. [17].

Despite these advantages, applying the WNM to high-order quantum systems ($L \geq 6$ [18]) remains challenging, mainly due to the rapid growth of the algebraic order as the group dimension grows. For instance, the special unitary group $SU(N)$ has $L = N^2 - 1$ generators in its associated algebra $\mathfrak{su}(N)$. Consequently, the number of Wei-Norman equations required to determine the TEO grows substantially, eventually rendering the derivation and manipulation of these equations infeasible by hand. Furthermore, if one wishes to compute the TEO using a different ordering of exponentials or an alternative basis, the entire derivation must typically be redone from scratch [19]. When only an approximate evolution is required, methods such as the Suzuki-Trotter decomposition [20] or the Magnus expansion [3] are commonly employed. However, these approaches generally fail to fully exploit the underlying algebraic structure of the dynamics, limiting their ability to reveal scaling laws or to obtain closed-form analytical solutions.

To overcome these and other limitations involving high-dimensional applications, we introduce *syndyn*, a Python library that automates the WNM. To the best of our knowledge, no open-source tool currently offers this capability. In addition to computing the Wei-Norman equations, the library automates the evaluation of nested commutators and nested similarity transformations, enabling the calculation of time evolution of any smooth function of the generators in the Heisenberg picture.

This paper is organized as follows: In Sec. II, we present the WNM in detail, emphasizing key results relevant to its automation. We focus on the computation of the time derivative of a Lie group element, which is central to the Wei-Norman framework, and examine the impact of basis choice. Meanwhile, we analytically recover nontrivial results for quantum systems associated with the low-order Lie algebras $\mathfrak{su}(1, 1)$, $\mathfrak{su}(2)$, $\mathfrak{sl}(2)$, and $\mathfrak{so}(2, 1)$ [21,22], relevant to models including the time-dependent harmonic oscillator and qubit dynamics. We conclude with the derivation of two $SU(2)$ gates essential in a configuration for universal quantum computing. In Sec. III, we demonstrate *syndyn*'s power and versatility in dealing with high-order quantum systems by deriving the Wei-Norman equations for a system of two coupled time-dependent harmonic oscillators with time-dependent coupling [23]. This example also serves as a practical guide to using *syndyn*. In Sec. IV we specialize our results to $SU(N)$ using a generic Cartan-Weyl basis for its associated algebra, allowing us to use the library for arbitrary N . We then apply our results to $SU(3)$ and $SU(4)$, and complete the derivation of quantum gates for universal quantum computation. We close with conclusions and perspectives in Sec. V.

To improve readability, we present below an explicit list of the main results of our work.

Using the bases defined in Table I(a), we obtain

- (i) Wei-Norman equations for the Lie algebras $\mathfrak{su}(1, 1)$, $\mathfrak{su}(2)$, $\mathfrak{sl}(2)$, and $\mathfrak{so}(2, 1)$ [Eqs. (38)–(40)].
- (ii) Explicit TEO for $\mathfrak{su}(2)$ [Eq. (54)]. In the same subsection, we discuss why the Pauli basis cannot be used to obtain a factorized TEO within the WNM.
- (iii) BCH-like relations for the Lie algebras $\mathfrak{su}(1, 1)$, $\mathfrak{su}(2)$, $\mathfrak{so}(2, 1)$, and $\mathfrak{sl}(2)$ [Eqs. (A12)–(A14)].
- (iv) T gate derived from the explicit TEO of $\mathfrak{su}(2)$ [Eq. (58)].
- (v) Hadamard gate derived from the explicit TEO of $\mathfrak{su}(2)$ [Eq. (60)].

For a system of two coupled time-dependent harmonic oscillators with time-dependent coupling, we obtain

- (i) Wei-Norman equations [Eqs. (64)–(74)].

Using the Cartan-Weyl basis defined in Sec. IV A, we obtain

- (i) Wei-Norman equations and explicit TEO for $\mathfrak{su}(2)$ [Eqs. (79)–(82)].
- (ii) Wei-Norman equations and explicit TEO for $\mathfrak{su}(3)$ [Eqs. (91)–(99)]. In the same subsection, we discuss why the Gell-Mann matrices cannot be used as a basis to obtain a factorized TEO within the WNM.
- (iii) Wei-Norman equations and explicit TEO for $\mathfrak{su}(4)$ [Eqs. (E3)–(E18)].
- (iv) CNOT gate derived from the explicit TEO of $\mathfrak{su}(4)$ [Eq. (100)].

II. WEI-NORMAN METHOD AS AN ALGORITHM

Here, we present a comprehensive and detailed treatment of the WNM, highlighting key results that enable its automation. For simplicity, we refer to quantum systems whose Hamiltonians are Lie vectors as *quantum Lie systems* (QLSs), with the system's order determined by the number of generators, i.e., the algebraic order L .

To make the calculation of the time derivative of Eq. (5)—a central step in the Wei-Norman framework—more intuitive, we introduce a graphical representation of the structure constants arranged as a tensor. To illustrate the capabilities of the WNM, we analytically obtain similarity transformations, BCH-like relations, and the Wei-Norman equations for QLSs associated with the algebras $\mathfrak{su}(1, 1)$, $\mathfrak{su}(2)$, $\mathfrak{sl}(2)$, and $\mathfrak{so}(2, 1)$. We then use $\mathfrak{su}(2)$ as a case to study the usage of the WNM in a suitably chosen basis. Further, we show how the formalism can be used to construct the Hadamard and T gates, fundamental components in a configuration for universal quantum computing.

We emphasize that the objective of the WNM is to derive the Wei-Norman equations for QLSs whose Hamiltonians satisfy $[\hat{H}(t), \hat{H}(t')] \neq 0$ and involve at least two algebra generators ($L \geq 2$). In the case $L = 1$, Eqs. (4) and (5) coincide, implying $[\hat{H}(t), \hat{H}(t')] = 0$, and thus there is nothing to factorize. More generally, when a Hamiltonian with $L \geq 2$ commutes with itself at different times, the TEO can be computed straightforwardly as [1]

$$\hat{U}(t) = \exp\left(-i \int_0^t dt' \hat{H}(t')\right), \quad (6)$$

where we have set $t_0 = 0$.

A. Derivative of a factorized Lie group element

At the heart of the WNM is the fact that the derivative of an element of a Lie group in its factorized representation can be written as a Lie vector multiplied by the element itself. We show below that to obtain this vector it is necessary to compute the set of all possible similarity transformations between the generators. Furthermore, we show how these results can be obtained systematically.

Without any loss of generality, let us consider for the product in Eq. (5) the ordering

$$\hat{U} = e^{\Lambda_1 \hat{g}_1} e^{\Lambda_2 \hat{g}_2} \dots e^{\Lambda_L \hat{g}_L}, \quad (7)$$

where we have omitted the temporal dependence in the argument of the coefficients for simplicity of notation. We shall do that along the text whenever there is no risk of confusion. Using the product rule, and that $e^{\theta \hat{A}} e^{-\theta \hat{A}} = e^{-\theta \hat{A}} e^{\theta \hat{A}} = \hat{\mathbb{1}}$ for arbitrary $\hat{A}$ and $\theta \in \mathbb{C}$, it is straightforward to show that

$$\begin{aligned} \frac{d}{dt} \hat{U} = & (\dot{\Lambda}_1 e^{\Lambda_0 \hat{g}_0} \hat{g}_1 e^{-\Lambda_0 \hat{g}_0} \\ & + \dot{\Lambda}_2 e^{\Lambda_0 \hat{g}_0} e^{\Lambda_1 \hat{g}_1} \hat{g}_2 e^{-\Lambda_1 \hat{g}_1} e^{-\Lambda_0 \hat{g}_0} + \dots \\ & + \dot{\Lambda}_L e^{\Lambda_0 \hat{g}_0} \dots e^{\Lambda_{L-1} \hat{g}_{L-1}} \hat{g}_L e^{-\Lambda_{L-1} \hat{g}_{L-1}} \dots e^{-\Lambda_0 \hat{g}_0}) \hat{U}, \end{aligned} \quad (8)$$

where we defined $\Lambda_0 \equiv 1$ and $\hat{g}_0 \equiv \hat{\mathbb{1}}$, and the overdot indicates time derivative. Notice that, depending on the algebra, one of the generators may be the identity again. We next show

that the above sum of nested similarity transformations, inside the parentheses, is a Lie vector and calculate it explicitly.

1. Single similarity transformations

Initially, to calculate a *single similarity transformation* we can use the well-known BCH lemma [24], which for two arbitrary generators reads [1]

$$\begin{aligned} e^{\Lambda_i \hat{g}_i} \hat{g}_j e^{-\Lambda_i \hat{g}_i} = & \hat{g}_j + \Lambda_i [\hat{g}_i, \hat{g}_j] + \frac{(\Lambda_i)^2}{2!} [\hat{g}_i, [\hat{g}_i, \hat{g}_j]] \\ & + \frac{(\Lambda_i)^3}{3!} [\hat{g}_i, [\hat{g}_i, [\hat{g}_i, \hat{g}_j]]] + \dots \end{aligned} \quad (9)$$

The nested commutators above correspond to derivatives of the similarity transformation in Eq. (9) with respect to Λ_i , subsequently evaluated at $\Lambda_i = 0$ [25]. More specifically, if we introduce the linear operator $\text{ad}_{\hat{g}_i}$ as

$$(\text{ad}_{\hat{g}_i})^n \hat{g}_j \equiv \frac{d^n (e^{\Lambda_i \hat{g}_i} \hat{g}_j e^{-\Lambda_i \hat{g}_i})}{d(\Lambda_i)^n} \Big|_{\Lambda_i=0}, \quad (10)$$

with $(\text{ad}_{\hat{g}_i})^0 \hat{g}_j \equiv \hat{g}_j$, then we can write a single commutator as

$$(\text{ad}_{\hat{g}_i}) \hat{g}_j = [\hat{g}_i, \hat{g}_j], \quad (11)$$

and the powers of this operator will indicate the number of commutators that are nested. For instance,

$$(\text{ad}_{\hat{g}_i})^3 \hat{g}_j = [\hat{g}_i, [\hat{g}_i, [\hat{g}_i, \hat{g}_j]]]. \quad (12)$$

With this definition, the right-hand side of Eq. (9) is written more compactly,

$$e^{\Lambda_i \hat{g}_i} \hat{g}_j e^{-\Lambda_i \hat{g}_i} = (e^{\Lambda_i \text{ad}_{\hat{g}_i}}) \hat{g}_j. \quad (13)$$

Using the definition in Eq. (2), nested commutators can be written in terms of the structure constants as

$$\begin{aligned} (\text{ad}_{\hat{g}_i}) \hat{g}_j &= \sum_{l=1}^L \gamma_{ij}^l \hat{g}_l, \\ (\text{ad}_{\hat{g}_i})^2 \hat{g}_j &= \sum_{l=1}^L \sum_{s_1=1}^L \gamma_{ij}^{s_1} \gamma_{is_1}^l \hat{g}_l, \\ (\text{ad}_{\hat{g}_i})^3 \hat{g}_j &= \sum_{l=1}^L \sum_{s_1=1}^L \sum_{s_2=1}^L \gamma_{ij}^{s_1} \gamma_{is_1}^{s_2} \gamma_{is_2}^l \hat{g}_l, \end{aligned} \quad (14)$$

and so on. Notice that the structure constants satisfy $\gamma_{ij}^l = -\gamma_{ji}^l$, $\gamma_{ij}^l = 0$ for $i = j$, and $\gamma_{0j}^l = \gamma_{i0}^l = 0$, as the identity commutes with all the elements of the group. Also noticed that nested commutators are also Lie vectors.

To help with the calculation of nested commutators, we introduce the *structure tensor* γ , whose components are the structure constants γ_{ij}^l . The two covariant components identify the generators involved in the commutator, and the contravariant component indicates which generator of the resulting Lie vector it is the coefficient of. This tensor can be viewed as the ordered set of all Lie vectors resulting from the commutators between all the generators of the algebra, and can be represented as in Fig. 1, where each vertical layer is a skew-symmetric matrix corresponding to $l = 1, 2, \dots, L$, with i indexing its rows and j its columns.

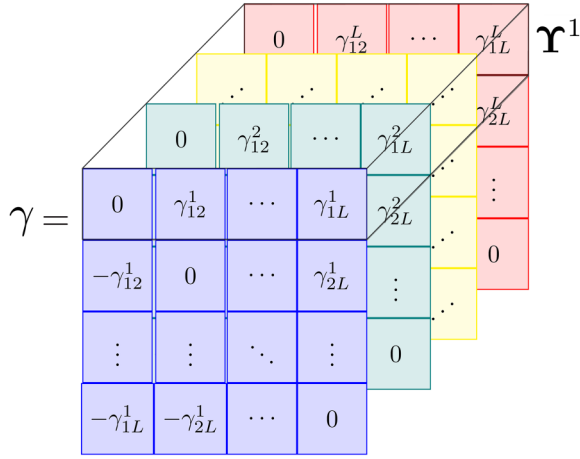


FIG. 1. Representation of the structure tensor γ and its top transverse matrix Υ^1 .

Thus, γ contains all the relevant information about the algebra for a particular ordering of the exponential generators in the factorized representation. Changing this ordering involves reordering the entries of γ . Each value of $i = 1, 2, \dots, L$ determines an $L \times L$ (transverse) matrix in γ , formed by the set of Lie vectors resulting from the commutator between the i th generator and each of the generators of the algebra, including itself. We define the set of these matrices as $\{\Upsilon^i\}_{i=1}^L$, and choose, for future convenience, to index their rows by j and their columns by l . This is a slightly different but equivalent definition to that in Ref. [26]. The j th row of Υ^i contains the coefficients of the Lie vector resulting from the commutator between the i th and the j th generator of the algebra, and therefore is null when $j = i$. For instance, the matrix corresponding to Υ^1 is the one at the top of γ , as indicated in Fig. 1, having its first row null. Using this definition it can be inferred from Eqs. (14) that nested commutators are equivalent to the powers of Υ^i [27]. More specifically [28],

$$(\text{ad } \hat{g}_i)^m \hat{g}_j = \sum_{l=1}^L ((\Upsilon^i)^m)_{jl} \hat{g}_l. \quad (15)$$

This implies that we can write a similarity transformation as

$$(e^{\Lambda_i \text{ad } \hat{g}_i}) \hat{g}_j = \sum_{l=1}^L b_{jl}^i \hat{g}_l, \quad (16)$$

where we defined

$$b_{jl}^i \equiv \sum_{m=0}^{\infty} \frac{(\Lambda_i)^m}{m!} ((\Upsilon^i)^m)_{jl} = (e^{\Lambda_i \Upsilon^i})_{jl}, \quad (17)$$

for $i, j \geq 1$. We also define $b_{jl}^0 \equiv \delta_{jl}$ and $(e^{\Lambda_i \text{ad } \hat{g}_i}) \hat{g}_0 = \hat{g}_0$, where δ is the Kronecker delta. It must be noted that b_{jl}^i are the components of a tensor made of all Lie vectors resulting from all possible similarity transformations between the generators of the algebra, with a coefficient multiplying the generator in the exponential. We will refer to this tensor as $\mathbf{b}$, with component matrices $\{\mathbf{b}^i\}_{i=1}^L$, so that

$$\mathbf{b}^i = e^{\Lambda_i \Upsilon^i}, \quad (18)$$

with $\mathbf{b}^0$ the $L \times L$ identity matrix. The calculation of a particular similarity transformation is now reduced to selecting a row (j index) of a $\mathbf{b}$ matrix (i index), and identify its entries (l index) with the components of the resulting Lie vector. We shall call $\mathbf{b}$ as the BCH tensor, and $\mathbf{b}^i$ as the BCH matrices.

Let us stress the importance of the BCH tensor by calculating it for the Lie algebras $\mathfrak{su}(1, 1)$, $\mathfrak{su}(2)$, $\mathfrak{sl}(2)$, and $\mathfrak{so}(2, 1)$. There are many prominent physical quantum systems associated with these algebras of order three. For instance, a time-dependent qubit, belonging to the Lie algebra $\mathfrak{su}(2)$ [30], or a time-dependent quantum harmonic oscillator, belonging to the Lie algebra $\mathfrak{su}(1, 1)$ [31–33], being fundamental systems of great importance in many fields of physics [14,15,34–45], allowing for the exploration of effects such as squeezing [46,47] and spin inversion [48,49], both crucial for advancing quantum technologies. Even some models for coupled harmonic oscillators, coupled spins or coupled two-photon lasers can also be associated with these algebras [50]. Moreover, they can be parameterized to be worked out within an unified way, as we show in Table I. There, the commutators between the algebra generators are given in Table I(a), while parameters ϵ and ν allow to specify the algebra, as indicated in Table I(b). Table I(a) is therefore a compact way of writing the structure tensor, and will be called a *structure table*. Recall that transverse matrices Υ^i can be rapidly deduced from the commutators between the generators. Using Table I(a), these matrices are calculated directly to be

$$\begin{aligned} \Upsilon^1 &= \begin{bmatrix} 0 & 0 & 0 \\ -\nu & 0 & 0 \\ 0 & -2\epsilon & 0 \end{bmatrix}, \quad \Upsilon^2 = \begin{bmatrix} \nu & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & -\nu \end{bmatrix}, \\ \Upsilon^3 &= \begin{bmatrix} 0 & 2\epsilon & 0 \\ 0 & 0 & \nu \\ 0 & 0 & 0 \end{bmatrix}. \end{aligned} \quad (19)$$

It can be noted that matrices Υ^1 and Υ^3 are nilpotent of degree 3, once $(\Upsilon^1)^3 = (\Upsilon^3)^3 = \mathbf{0}$. Thus, to calculate the BCH matrices we only need to compute up to their squares

$$(\Upsilon^1)^2 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 2\epsilon\nu & 0 & 0 \end{bmatrix}, \quad (\Upsilon^3)^2 = \begin{bmatrix} 0 & 0 & 2\epsilon\nu \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}. \quad (20)$$

TABLE I. (a) Structure table for the Lie algebras $\mathfrak{su}(1, 1)$, $\mathfrak{su}(2)$, $\mathfrak{sl}(2)$, and $\mathfrak{so}(2, 1)$. Each entry of its inner 3×3 (structure) matrix corresponds to the Lie vector resulting from the ordered commutator between the generator at the left with the one at the top. Parameters ϵ and ν allow to specify the algebra as indicated in part (b).

(a)				(b)		
	$\hat{g}_1$	$\hat{g}_2$	$\hat{g}_3$	Lie algebra	ϵ	ν
$\hat{g}_1$	0	$-\nu \hat{g}_1$	$-2\epsilon \hat{g}_2$	$\mathfrak{su}(1, 1)$	1	1
$\hat{g}_2$	$\nu \hat{g}_1$	0	$-\nu \hat{g}_3$	$\mathfrak{su}(2)$	-1	1
$\hat{g}_3$	$2\epsilon \hat{g}_2$	$\nu \hat{g}_3$	0	$\mathfrak{sl}(2)$	-1/2	2
				$\mathfrak{so}(2, 1)$	$i/2$	i

On the other hand, Υ^2 is diagonal, and then is direct that

$$(\Upsilon^2)^m = \begin{bmatrix} (\nu)^m & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & (-\nu)^m \end{bmatrix}. \quad (21)$$

Finally, for any transverse matrix we have

$$(\Upsilon^i)^0 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & 1 \end{bmatrix}. \quad (22)$$

Using the above results and Eq. (18), we can calculate the BCH matrices as

$$\begin{aligned} \mathbf{b}^1 &= \begin{bmatrix} 1 & 0 & 0 \\ -\nu\Lambda_1 & 1 & 0 \\ \epsilon\nu(\Lambda_1)^2 & -2\epsilon\Lambda_1 & 1 \end{bmatrix}, \\ \mathbf{b}^2 &= \begin{bmatrix} e^{\nu\Lambda_2} & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & e^{-\nu\Lambda_2} \end{bmatrix}, \\ \mathbf{b}^3 &= \begin{bmatrix} 1 & 2\epsilon\Lambda_3 & \epsilon\nu(\Lambda_3)^2 \\ 0 & 1 & \nu\Lambda_3 \\ 0 & 0 & 1 \end{bmatrix}. \end{aligned} \quad (23)$$

As mentioned before, any similarity transformation in the algebra can be now easily calculated. For instance,

$$(e^{\Lambda_1 \text{ad} \hat{g}_1})\hat{g}_2 = \sum_{l=1}^L b_{2l}^1 \hat{g}_l = -\nu\Lambda_1 \hat{g}_1 + \hat{g}_2, \quad (24)$$

corresponding with the second row of $\mathbf{b}^1$ in Eq. (23). It is important to note that the BCH matrices must be explicitly computed to proceed with the subsequent steps of the method, and their computation depends on the specific algebra under consideration. Fortunately, there are many different techniques and frameworks to calculate matrix exponentials (see, for instance, Refs. [26,51–53]). For our numerical calculations we use sympy [54]. Moreover, by using a suitable basis [6,7], as we will show in Sec. II B 1, we can expect transverse matrices to be nilpotent or diagonal, simplifying the calculation of their powers as in the above example.

2. Nested similarity transformations

Now that we have shown how the set of single similarity transformations can be computed, we can move on to compute *nested similarity transformations*. Using Eq. (16), it is straightforward to show that r ordered [as in Eq. (7)] nested similarity transformations can be expressed as

$$\prod_{i=0}^r (e^{\Lambda_i \text{ad} \hat{g}_i})\hat{g}_j = \sum_{l=1}^L \beta_{jl}^r \hat{g}_l, \quad (25)$$

for $r \in \{1, 2, \dots, L\}$, where we have defined

$$\begin{aligned} \beta_{jl}^r &\equiv \sum_{s_1=1}^L \sum_{s_2=1}^L \cdots \sum_{s_{r-1}=1}^L b_{js_1}^r b_{s_1 s_2}^{r-1} \cdots b_{s_{r-2} s_{r-1}}^2 b_{s_{r-1} l}^1 \\ &= (\mathbf{b}^r \mathbf{b}^{r-1} \cdots \mathbf{b}^1)_{jl}. \end{aligned} \quad (26)$$

Additionally, any sequence of nested similarity transformations, not necessarily ordered, can be also computed by simply multiplying specific BCH matrices.

Coming back to Eq. (8), using Eq. (13) it can be put in the form

$$\frac{d}{dt}\hat{U} = \sum_{n=1}^L \dot{\Lambda}_n \left(\prod_{i=0}^{n-1} (e^{\Lambda_i \text{ad} \hat{g}_i})\hat{g}_n \right) \hat{U}. \quad (27)$$

The expression within the parentheses with the help of Eq. (25) becomes

$$\prod_{i=0}^{n-1} (e^{\Lambda_i \text{ad} \hat{g}_i})\hat{g}_n = \sum_{l=1}^L \xi_{nl} \hat{g}_l, \quad (28)$$

where we defined $\xi_{nl} \equiv \beta_{nl}^{n-1}$ as the elements of the matrix ξ , namely,

$$\xi_{nl} = (\mathbf{b}^{n-1} \mathbf{b}^{n-2} \cdots \mathbf{b}^1)_{nl}. \quad (29)$$

Notice that the n th row of ξ equals the n th row of the resulting ordered product of the first $n-1$ BCH matrices. We shall call ξ the *coupling matrix*, for reasons that will become clear later. Using the above results, we can finally write Eq. (27) as

$$\frac{d}{dt}\hat{U} = \left(\sum_{l=1}^L \sum_{n=1}^L \dot{\Lambda}_n \xi_{nl} \hat{g}_l \right) \hat{U}, \quad (30)$$

and thus we have shown that the derivative of a Lie element in the factorized representation is given by an explicit Lie vector (inside the parenthesis) times the element itself.

For clarity, let us calculate the coupling matrix for the low-order Lie algebras we have been working with. From Eqs. (23) and (29), the following results are straightforward: The first row of ξ is the same as the first row of $\mathbf{b}^0$, that is, the first row of the identity; the second row of ξ is the same as the second row of $\mathbf{b}^1$; and the third row of ξ is the same as the third row of the resulting matrix $\mathbf{b}^2 \mathbf{b}^1$, a product that can be calculated using Eqs. (23) as

$$\mathbf{b}^2 \mathbf{b}^1 = \begin{bmatrix} e^{\nu\Lambda_2} & 0 & 0 \\ -\nu\Lambda_1 & 1 & 0 \\ \epsilon\nu\frac{(\Lambda_1)^2}{e^{\nu\Lambda_2}} & -2\epsilon\frac{\Lambda_1}{e^{\nu\Lambda_2}} & \frac{1}{e^{\nu\Lambda_2}} \end{bmatrix}. \quad (31)$$

Using the above results, we finally obtain

$$\xi = \begin{bmatrix} 1 & 0 & 0 \\ -\nu\Lambda_1 & 1 & 0 \\ \epsilon\nu\frac{(\Lambda_1)^2}{e^{\nu\Lambda_2}} & -2\epsilon\frac{\Lambda_1}{e^{\nu\Lambda_2}} & \frac{1}{e^{\nu\Lambda_2}} \end{bmatrix}. \quad (32)$$

Notice that the determinant of the above coupling matrix is $\det|\xi| = \exp(-\nu\Lambda_2)$, and it vanishes in the limit $\Lambda_2 \rightarrow \infty$. Next, we will see the importance of calculating the determinant of the coupling matrix and the role of the basis in the usefulness of the method.

B. TEO coefficients and an appropriated basis

In the previous section, we calculated the specific Lie vector that determines the ratio between the derivative of a Lie element in the factorized representation and the element itself. For QLSs, such a vector must be proportional to the

Hamiltonian. More specifically, using Eqs. (1) and (30) in Eq. (3), we obtain

$$\sum_{l=1}^L \eta_l \hat{g}_l = i \sum_{l=1}^L \sum_{n=1}^L \dot{\Lambda}_n \xi_{nl} \hat{g}_l, \quad (33)$$

and considering that the generators are linearly independent, we must have

$$\eta_l = i \sum_{n=1}^L \dot{\Lambda}_n \xi_{nl}, \quad (34)$$

which is a set of time-dependent coupled differential equations with the initial conditions $\{\Lambda_n = 0\}_{n=1}^L$ at $t = 0$. It can be written in matrix form as

$$\underbrace{\begin{bmatrix} \eta_1(t) \\ \eta_2(t) \\ \vdots \\ \eta_L(t) \end{bmatrix}}_{\boldsymbol{\eta}} = i \underbrace{\begin{bmatrix} \xi_{11} & \xi_{21} & \cdots & \xi_{L1} \\ \xi_{12} & \xi_{22} & \cdots & \xi_{L2} \\ \vdots & \vdots & \ddots & \vdots \\ \xi_{1L} & \xi_{2L} & \cdots & \xi_{LL} \end{bmatrix}}_{\boldsymbol{\xi}^T} \underbrace{\begin{bmatrix} \dot{\Lambda}_1 \\ \dot{\Lambda}_2 \\ \vdots \\ \dot{\Lambda}_L \end{bmatrix}}_{\dot{\boldsymbol{\Lambda}}}, \quad (35)$$

where T indicates the transpose and we recall that all the explicit time-dependence is in $\boldsymbol{\eta}$. Now, it is clear why we called $\boldsymbol{\xi}$ the coupling matrix. The global validity of the above equations is only possible when $\boldsymbol{\xi}$ is invertible for all time, a condition verifiable by calculating its determinant $\det |\boldsymbol{\xi}|$. Nevertheless, there is always a local solution, as there exist at least a neighborhood of $t = 0$ where $\boldsymbol{\xi}$ is invertible. This is consequence from $\det |\boldsymbol{\xi}|$ be an analytic function of the TEO coefficients and $\det |\boldsymbol{\xi}(t \rightarrow 0)| = 1$ [5].

Considering $\det |\boldsymbol{\xi}| \neq 0$, we can go ahead and multiply by the inverse of $\boldsymbol{\xi}^T$ from the left in the above equation to obtain

$$(\boldsymbol{\xi}^T)^{-1} \boldsymbol{\eta} = i \dot{\boldsymbol{\Lambda}}, \quad (36)$$

which is a system of block-decoupled [55] nonlinear differential equations known as the *Wei-Norman equations*. On the other hand, if there are singularities in the coupling matrix, i.e., values of the TEO coefficients for which $\det |\boldsymbol{\xi}| = 0$, the global validity of the solution is not guaranteed. In fact, as mentioned by Wei and Norman, to find a global solution the choice of basis is critical [5]. In the next subsection, we will return to this important point related to the usefulness of the method. For now, as an example with a global solution let us obtain the Wei-Norman equations for the four algebras we have been working on.

Using Eq. (32) together with Eq. (35), we obtain directly

$$\begin{bmatrix} \eta_1(t) \\ \eta_2(t) \\ \eta_3(t) \end{bmatrix} = i \begin{bmatrix} 1 & -v\Lambda_1 & \epsilon v \frac{(\Lambda_1)^2}{e^{v\Lambda_2}} \\ 0 & 1 & -2\epsilon \frac{\Lambda_1}{e^{v\Lambda_2}} \\ 0 & 0 & \frac{1}{e^{v\Lambda_2}} \end{bmatrix} \begin{bmatrix} \dot{\Lambda}_1 \\ \dot{\Lambda}_2 \\ \dot{\Lambda}_3 \end{bmatrix}. \quad (37)$$

As $\det |\boldsymbol{\xi}^T| = \det |\boldsymbol{\xi}| = \exp(-v\Lambda_2)$, one can invert $\boldsymbol{\xi}^T$ for all times except when $\Lambda_2 \rightarrow \infty$. Multiplying by $(\boldsymbol{\xi}^T)^{-1}$ from the left in the above equation and then performing the matrix

TABLE II. Structure table for the Lie algebra $\mathfrak{su}(2)$ in the Pauli basis.

	$\hat{\underline{g}}_1$	$\hat{\underline{g}}_2$	$\hat{\underline{g}}_3$
$\hat{\underline{g}}_1$	0	$-\hat{\underline{g}}_3$	$\hat{\underline{g}}_2$
$\hat{\underline{g}}_2$	$\hat{\underline{g}}_3$	0	$-\hat{\underline{g}}_1$
$\hat{\underline{g}}_3$	$-\hat{\underline{g}}_2$	$\hat{\underline{g}}_1$	0

product, we obtain the Wei-Norman equations of the system

$$\dot{\Lambda}_1 + \epsilon v (i\eta_3)(\Lambda_1)^2 + v(i\eta_2)\Lambda_1 + i\eta_1 = 0, \quad (38)$$

$$\dot{\Lambda}_2 + 2\epsilon(i\eta_3)\Lambda_1 + i\eta_2 = 0, \quad (39)$$

$$\dot{\Lambda}_3 + (i\eta_3)e^{v\Lambda_2} = 0. \quad (40)$$

Equation (38) represents four families of time-dependent complex Riccati equations, each associated with one of the Lie algebras we are considering. The other two equations are solved by quadrature after solving this previous one. The above results are in agreement with the differential equations obtained for $\mathfrak{su}(1,1)$, $\mathfrak{su}(2)$, and $\mathfrak{so}(2,1)$ QLSs in Ref. [22] [see Eqs. (8)–(10) there, taking into account that they considered $\ln \Lambda_2$ as the coefficient of $\hat{\underline{g}}_2$], and such for the $\mathfrak{sl}(2)$ QLS established in Ref. [21] [see Eqs. (54)–(56) there].

Although the general time-dependent complex Riccati equation has no analytical solution for the general time-dependent case [56], Eq. (38) can be solved recursively and implemented numerically efficiently using the composition property of the TEO and the analytical solution of the time-independent case [22]. In Appendix A, we use this solution of the time-independent case to obtain BCH-like relations for the above algebras, while showing how the Wei-Norman factorization is carried out. In particular, BCH-like relations of $\mathfrak{su}(2)$ will be used to calculate two important one-qubit gates in Sec. II C as a confirmation of our results. Let us now explore the issue of the choice of basis to obtain a global solution.

1. Decoupling in an appropriate basis

To illustrate the impact of choosing an appropriate basis on the utility of the WNM, let us apply the method for $\mathfrak{su}(2)$ QLSs but this time using the Pauli matrices as a basis, which is the usual basis for working with qubits. More specifically, we consider the generators: $\hat{\underline{g}}_j = \frac{i}{2}\hat{\sigma}_j$ ($j = 1, 2, 3$), where

$$\hat{\sigma}_1 = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}, \quad \hat{\sigma}_2 = \begin{bmatrix} 0 & -i \\ i & 0 \end{bmatrix}, \quad \hat{\sigma}_3 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix} \quad (41)$$

are the Pauli matrices. The idea is to obtain the Wei-Norman equations for the coefficients of the TEO

$$\hat{U} = e^{\Theta_1 \hat{\underline{g}}_1} e^{\Theta_2 \hat{\underline{g}}_2} e^{\Theta_3 \hat{\underline{g}}_3}, \quad (42)$$

corresponding to a given Hamiltonian in this basis

$$\hat{H} = \varsigma_1 \hat{\underline{g}}_1 + \varsigma_2 \hat{\underline{g}}_2 + \varsigma_3 \hat{\underline{g}}_3. \quad (43)$$

The well-known commutation relations for Pauli matrices $[\sigma_j, \sigma_k] = 2i \sum_{l=1}^3 \epsilon_{jkl} \sigma_l$, where ϵ_{jkl} is the Levi-Civita symbol, allow us to build the structure table of the system (see Table II), and with it the structure tensor. From the latter, one

can calculate the corresponding coupling matrix, which is

$$\xi_P = \begin{bmatrix} 1 & 0 & 0 \\ 0 & \cos(\Theta_1) & -\sin(\Theta_1) \\ -\sin(\Theta_2) & \sin(\Theta_1)\cos(\Theta_2) & \cos(\Theta_1)\cos(\Theta_2) \end{bmatrix}. \quad (44)$$

Its determinant is given by $\det|\xi_P| = \cos(\Theta_2)$, so it has singularities at $\Theta_2 = \pi(n + 1/2)$, with $n \in \mathbb{Z}$. The coupling matrix (as well as its transpose) is therefore noninvertible for all times, and then this solution is not global. Although the inverse of the coupling matrix can be computed as

$$\xi_P^{-1} = \begin{bmatrix} 1 & 0 & 0 \\ \sin(\Theta_1)\tan(\Theta_2) & \cos(\Theta_1) & \sin(\Theta_1)\sec(\Theta_2) \\ \cos(\Theta_1)\tan(\Theta_2) & -\sin(\Theta_1) & \cos(\Theta_1)\sec(\Theta_2) \end{bmatrix}, \quad (45)$$

it does exist for all values of Θ_l except in the mentioned periodic singularities [57]. Besides, using the above equation and Eq. (36) we can calculate the corresponding Wei-Norman equations

$$\dot{\Theta}_1 + i \tan(\Theta_2)[\zeta_2 \sin(\Theta_1) + \zeta_3 \cos(\Theta_1)] + i\zeta_1 = 0, \quad (46)$$

$$\dot{\Theta}_2 + i[\zeta_2 \cos(\Theta_1) - \zeta_3 \sin(\Theta_1)] = 0, \quad (47)$$

$$\dot{\Theta}_3 + i \sec(\Theta_2)[\zeta_2 \sin(\Theta_1) + \zeta_3 \cos(\Theta_1)] = 0, \quad (48)$$

which remain coupled, and decoupling is not evident. Accordingly, we cannot obtain a global factorized version of the TEO, as in Eq. (42), using the Pauli matrices as the generators of $\mathfrak{su}(2)$, either starting with an unfactorized version of the TEO, or with a Hamiltonian in this basis. Nevertheless, a simple change of basis, as anticipated by Wei and Norman, can change everything. If we define ladder operators $\hat{g}_1 \equiv \hat{\sigma}_+ = \frac{1}{2}(\hat{\sigma}_1 + i\hat{\sigma}_2)$ and $\hat{g}_3 \equiv \hat{\sigma}_- = \frac{1}{2}(\hat{\sigma}_1 - i\hat{\sigma}_2)$, together with $\hat{g}_2 = \frac{1}{2}\hat{\sigma}_3$, then the algebra falls in the commutation relations for $\mathfrak{su}(2)$ given in Table I(a), and consequently the TEO can be expressed as

$$\hat{U} = e^{\Lambda_1 \hat{g}_1} e^{\Lambda_2 \hat{g}_2} e^{\Lambda_3 \hat{g}_3}, \quad (49)$$

where Λ_1 , Λ_2 , and Λ_3 are given by Eqs. (38)–(40) with $\epsilon = -1$ and $\nu = 1$. Further, we can calculate the TEO in terms of the Pauli matrices as follows. From their definitions and Eq. (41), it can be shown that $\hat{g}_1$ and $\hat{g}_3$ are nilpotent of degree 2, and therefore

$$e^{\Lambda_1 \hat{g}_1} = \hat{1} + \frac{\Lambda_1}{2}(\hat{\sigma}_1 + i\hat{\sigma}_2), \quad e^{\Lambda_3 \hat{g}_3} = \hat{1} + \frac{\Lambda_3}{2}(\hat{\sigma}_1 - i\hat{\sigma}_2). \quad (50)$$

On the other hand,

$$\begin{aligned} e^{\Lambda_2 \hat{g}_2} &= \sum_{n=0}^{\infty} \frac{1}{n!} \left(\frac{\Lambda_2}{2} \right)^n (\hat{\sigma}_3)^n \\ &= \cosh\left(\frac{\Lambda_2}{2}\right) \hat{1} + \sinh\left(\frac{\Lambda_2}{2}\right) \hat{\sigma}_3, \end{aligned} \quad (51)$$

where we have used that $\hat{\sigma}_i^2 = \hat{1}$ for $i = 1, 2, 3$. Substituting Eqs. (50) and (51) in Eq. (49), it is straightforward to show

that

$$\begin{aligned} \hat{U} &= \frac{e^{-\frac{\Lambda_2}{2}}}{2} ((e^{\Lambda_2} + 1 + \Lambda_1 \Lambda_3) \hat{1} + (\Lambda_1 + \Lambda_3) \hat{\sigma}_1 \\ &\quad + (\Lambda_1 - \Lambda_3) \hat{\sigma}_2 + (e^{\Lambda_2} - 1 + \Lambda_1 \Lambda_3) \hat{\sigma}_3), \end{aligned} \quad (52)$$

where we have used the well-known property of the Pauli matrices $\hat{\sigma}_j \hat{\sigma}_k = \delta_{jk} \hat{1} + i \sum_{l=1}^3 \epsilon_{jkl} \hat{\sigma}_l$.

Now, using Eq. (41) we can compute the TEO in its explicit matrix form as

$$\hat{U} = e^{-\frac{\Lambda_2}{2}} \begin{bmatrix} e^{\Lambda_2} + \Lambda_1 \Lambda_3 & \Lambda_1 \\ \Lambda_3 & 1 \end{bmatrix}, \quad (53)$$

which satisfies $\det|\hat{U}| = 1$. Notice that we could have computed the TEO in the above explicit form by directly substituting the generators in their matrix form into Eq. (49), then computing the matrix exponentials and finally doing their products. This direct way of computing the TEO will be used when working with higher-order algebras.

To guarantee the unitarity of the TEO in Eq. (53), we deduced the correspondent constraints in Appendix B. Writing the TEO coefficients in its polar form $\Lambda_1 = |\Lambda_1|e^{i\varphi_1}$, $\Lambda_2 = |\Lambda_2|e^{i\varphi_2}$, and $\Lambda_3 = |\Lambda_3|e^{i\varphi_3}$, the three constraints we obtained are $|\Lambda_3| = |\Lambda_1|$, $e^{\Re(\Lambda_2)} = 1 + |\Lambda_1|^2$, and $e^{i\Im(\Lambda_2)} = -e^{i(\varphi_1 + \varphi_3)}$. Using the above results, we can rewrite Eq. (53) as

$$\hat{U} = \frac{1}{\sqrt{1 + |\Lambda_1|^2}} \begin{bmatrix} -e^{i(\varphi_1 + \varphi_3)} & |\Lambda_1|e^{i\varphi_1} \\ |\Lambda_1|e^{i\varphi_3} & 1 \end{bmatrix}, \quad (54)$$

where we have disregarded the global phase. Notice the dependence on the three real parameters $|\Lambda_1|$, φ_1 , and φ_3 .

We have shown that changing the basis of the Pauli matrices to that of ladder operators yields an invertible coupling matrix for all times in $\mathfrak{su}(2)$ QLSs, allowing us to explicitly calculate the TEO. This has been possible because the bases we defined for the algebras in Table I(a) are Cartan-Weyl bases, and they naturally align with the root structure of the Lie algebra. Moreover, according to Wei and Norman (theorem 5 in Ref. [5]), when working with semisimple algebras, which have no nontrivial solvable ideals, this kind of decomposition serves to guarantee an invertible coupling matrix.

To construct a Cartan-Weyl basis (CWB) for a semisimple Lie algebra, one typically begins by selecting a Cartan subalgebra, which is a maximal abelian subalgebra. It is usual to choose diagonal matrices as its generators. The next step is to construct the associated root spaces, whose elements (equivalent to raising and lowering operators) are Lie vectors proportional to themselves when their commutators with the elements of the Cartan subalgebra are calculated. The proportionality constants are known as roots, and they come in pairs with opposite sign. In our previous example for $\mathfrak{su}(2)$, $\hat{g}_1$ and $\hat{g}_3$ can be identified with root vectors from different root spaces, with roots $+1$ and -1 , respectively, and $\hat{g}_2$ is the only element of the Cartan subalgebra. The above is not intended to be a formal definition but rather a practical description of a CWB. For an explicit definition we suggest Ref. [58], and for a formal definition we suggest Refs. [59,60]. In Sec. III, we build a generic CWB for the Lie algebra $\mathfrak{su}(N)$, going a bit deeper into this practical description.

A CWB decomposition within the WNM provides both theoretical and computational advantages for classical simple

algebras [6,7]. From the theoretical side, the CWB extends the familiar ladder-operator formalism of the harmonic oscillator and spin-1/2 systems to arbitrary-dimensional classical Lie algebras. It decomposes the algebra into commuting Cartan generators and pairs of raising and lowering operators associated with each root, naturally organizing its structure in terms of transitions between weight states. From the computational side, all matrix exponentials required to compute the BCH matrices [see Eq. (18)] reduce to polynomials, which are straightforward to evaluate. This simplification occurs because the matrices associated with each root subspace are strictly upper or lower triangular, and thus nilpotent, while the generators belonging to the Cartan subalgebra are block diagonal. This behavior is clearly exhibited in Eqs. (19)–(21), for instance. Furthermore, when the TEO is expressed as the exponential of the positive (or negative) root, followed by the exponential of the Cartan subalgebra, and then the exponential of the negative (or positive) root, the Wei-Norman equations acquire the form of a system of block-decoupled nonlinear differential equations exhibiting a hierarchical structure of coupled Riccati equations in the case of high-order QLSs, or a single Riccati equation for low-order QLSs as in Eq. (38). We will observe this structure clearly in the following sections, when working with time-dependent coupled harmonic oscillators in Sec. III A and $\mathfrak{su}(N)$ in Sec. IV.

Next, we will evaluate and use our solution of Eq. (54) in the construction of two important one-qubit quantum gates.

C. H and T single-qubit gates

Universal quantum computation can be achieved using the one-qubit Hadamard and T gates, along with the two-qubit CNOT gate [61–63]. Here we test and use our solution in Eq. (54) to demonstrate the construction of the one-qubit mentioned gates.

We consider the time-independent Hamiltonian of a qubit driven by a laser (or a microwave for nuclear magnetic resonance) that has detuning Δ and intensity Ω

$$\hat{H} = \frac{\Omega}{2}\hat{\sigma}_1 + \Delta\frac{\hat{\sigma}_3}{2}. \quad (55)$$

First, recalling that $\hat{g}_1 = \frac{1}{2}(\hat{\sigma}_1 + i\hat{\sigma}_2)$, $\hat{g}_3 = \frac{1}{2}(\hat{\sigma}_1 - i\hat{\sigma}_2)$, and $\hat{g}_2 = \frac{1}{2}\hat{\sigma}_3$, we can write $\hat{\sigma}_1 = \hat{g}_1 + \hat{g}_3$ and put the Hamiltonian in the CWB of the previous section, so we can identify $\eta_1 = \eta_3 = \Omega/2$ and $\eta_2 = \Delta$. The qubit state evolves with the differential equation

$$\frac{d}{dt}\langle\hat{\sigma}\rangle = \mathbf{P} \times \langle\hat{\sigma}\rangle, \quad (56)$$

where $\mathbf{P} = (\Omega, 0, \Delta)^T$ is a precession vector, and $\langle\hat{\sigma}\rangle = (\langle\hat{\sigma}_1\rangle, \langle\hat{\sigma}_2\rangle, \langle\hat{\sigma}_3\rangle)^T$ is the Bloch vector [64], with $\langle\hat{\sigma}_i\rangle$ indicating the expected value of operator $\hat{\sigma}_i$. The T gate is realized by setting the precession vector pointing to the $+z$ direction in the Bloch sphere, and carrying out $1/8$ of a revolution. This is done by setting $\Omega = 0$. Further, recalling that the Hamiltonian in Eq. (55) is time-independent, the TEO can directly be calculated in the unfactorized representation of Eq. (4), allowing us to identify $\lambda_j = -i\eta_j t$, with $j = 1, 2, 3$. Using the BCH-like equations [Eqs. (A12)–(A14)], we get for the T gate

$\Lambda_1 = \Lambda_3 = 0$ and $\Lambda_2 = -it\Delta$, where t is the application time of the Hamiltonian, which should be sufficient to carry out one eighth of a revolution, i.e. $t = \pi/(4\Delta)$. Accordingly, using the unitarity constraints we get $|\Lambda_1| = |\Lambda_3| = \Re(\Lambda_2) = 0$ and $\Im(\Lambda_2) = -\pi/4$. Replacing these results in Eq. (54) shows that

$$\hat{U}_T = \begin{bmatrix} e^{-i\pi/4} & 0 \\ 0 & 1 \end{bmatrix}, \quad (57)$$

or, multiplying by the global phase $e^{i\pi/4}$, we obtain the canonical representation of the T gate:

$$\hat{U}_T = \begin{bmatrix} 1 & 0 \\ 0 & e^{i\pi/4} \end{bmatrix}. \quad (58)$$

The Hadamard gate H is realized by pointing the precession vector to the direction $(1,0,1)$, i.e., by setting $\Omega = \Delta$. This implies that $\lambda_1 = \lambda_3 = \lambda_2/2 := \lambda$. Replacing this in Eqs. (A12)–(A14) we obtain

$$\Lambda_1 = \frac{\sinh(\sqrt{2}\lambda)}{\sqrt{2}\cosh(\sqrt{2}\lambda) - \sinh(\sqrt{2}\lambda)}. \quad (59)$$

The application time is $t = \pi/(\sqrt{2}\Delta)$, meaning that $\lambda = -i\pi/(2\sqrt{2})$, which implies that $\Lambda_1 = \Lambda_3 = -1$ and $\Lambda_2 = \ln 2 - i\pi$. Accordingly, using the unitarity constraints we get $|\Lambda_1| = |\Lambda_3| = 1$ and $e^{i\varphi_1} = e^{i\varphi_3} = -1$. Replacing these results in Eq. (54) shows that

$$\hat{U}_H = \frac{1}{\sqrt{2}} \begin{bmatrix} -1 & -1 \\ -1 & 1 \end{bmatrix}, \quad (60)$$

which is the Hadamard matrix up to a global phase of $e^{i\pi}$. The obtainment of the above quantum gates can be understood as a confirmation of our results.

Although recovering the $\hat{\sigma}_3$ gate using Eq. (54) is straightforward, we note that for the other two individual gates, those proportional to $\hat{\sigma}_1$ and $\hat{\sigma}_2$, it requires considering the limit in which the coupling matrix is singular $\Lambda_2 \rightarrow \infty$ [see Eq. (37)]. However, this limit can be carefully computed and the mentioned gates can be recovered, as we show in the Appendix C. Moreover, it is well-known that specific products of the Hadamard and T gates give rise to these gates. Finally, recall that the computation of time-dependent single gates is direct using Eq. (6).

III. Symdyn

In this section, we demonstrate the power and flexibility of our library in dealing with high-order QLSs, by getting key results of the WNM for a quantum system of two coupled time-dependent harmonic oscillators with time-dependent coupling. This example also serves to illustrate the practical use of symdyn. We point out that all analytical results of the previous section were computationally recovered using the library, validating it for low-order algebras. The reader can find setting up instructions and examples, including those contained in this work, in our repository at [65].

Let us begin by summarizing the key results of the WNM, corresponding to the shadow boxes of the flowchart in Fig. 2. The transparent ones indicate the interaction between the user and symdyn. As it can be seen, sometimes the user must

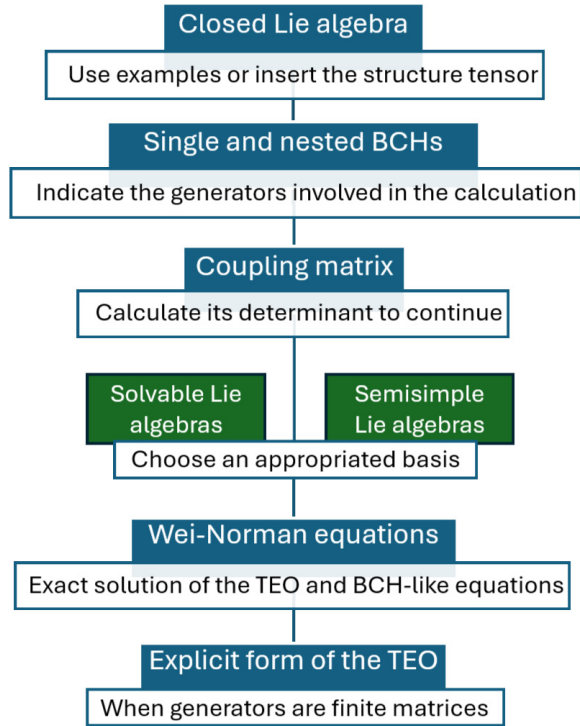


FIG. 2. Flowchart of the WNM.

specify the objects involved in the calculation to obtain the desired information, and other times only needs to request the information. Be aware that the last two steps of the flowchart are only possible depending on the chosen basis of the Lie Algebra with which you are working.

Symdyn's main class is called `Algebra`, which can be imported in Python as follows:

```
from symdyn.algebra import Algebra
```

To instantiate an `Algebra`, a structure tensor object must be passed. The user will have two ways to define the structure tensor. In case the generators are finite matrices, the user can input them one by one, and ask symdyn to compute the corresponding commutators to build γ for a specific ordering of the basis. We will explore this option for $SU(3)$ in Sec. IV. The other option is to input the structure tensor γ directly, as we show in the following example.

A. Two time-dependent coupled harmonic oscillators

The quantum harmonic oscillator is one of the most important systems in quantum physics. As already mentioned, its time-dependent version allows the study of squeezing, which consists in reducing the uncertainty of one of the quadratures of the oscillator below the value for coherent states (for the other quadrature, the uncertainty increases accordingly) by using the nonadiabatic variation of the oscillator parameters (mass and frequency). This effect is relevant in many fields of physics [46,66–86], and useful to enhancing the sensitivity of several systems [87,88], including telecommunications [89–91] and the detection rate at LIGO [92–94].

Our first high-order QLS consist of two time-dependent harmonic oscillators coupled with a time-dependent coupling. The most general case involves a total of 15 generators and

all time-dependent physical parameters [95]. A simplified version of this problem was recently solved using the WNM together with a numerical approach by Ramos-Prieto *et al.* [23], in the context of effective approaches to the dynamical Casimir effect. The system consists of a one-dimensional cavity with a moving mirror in one boundary and two coupled modes of the electromagnetic field inside [96]. They reduced the number of generators to 11 by ignoring terms proportional to single ladder operators of each mode. In our development, we hold this reduction for the simplicity of the expressions. The effective Hamiltonian of the system can be written as

$$\hat{H}(t) = \sum_{l=1}^{11} \eta_l(t) \hat{g}_l, \quad (61)$$

where the coefficients are functions of the physical parameters of the system and the generators are bilinear products of ladder operators of each mode (see Appendix D), forming a CWB for the system.

Our first step is to calculate the structure tensor, which can be deduced from the structure table of the system (Table III). Notice that some commutators are proportional to a linear combination of two or more generators, unlike in our previous examples where they were proportional to a single generator. To encode γ in symdyn, one must initially define the algebraic order L to create a zero-filled $L \times L \times L$ tensor. For our current system $L = 11$, and the code to generate it is

```
import numpy as np
gamma = np.zeros((11, 11, 11))
```

We then proceed to indicate the nonzero entries of each transverse matrix. For instance, the top transverse matrix, corresponding to the first row of the structure matrix in Table III, is encoded as

```
gamma[0, 4, 2] = -1
gamma[0, 5, 0] = -2
gamma[0, 7, 5] = -4
gamma[0, 7, 10] = -2
gamma[0, 9, 3] = -2
```

where it must be noticed that Python counts elements of collections from zero. In this way, the first line in the above code indicates that the entry correspondent to the commutator between $\hat{g}_1$ and $\hat{g}_5$ (first two indexes) is proportional to $\hat{g}_3$ (third index), with coefficient of proportionality -1 . In case a commutator is proportional to more than one generator, lines (with the same first two indexes) must be written specifying each generator with its proportionality coefficient. For example, in the above code the third and fourth lines indicate that the commutator between $\hat{g}_1$ and $\hat{g}_8$ is given by the linear combination of the generators $\hat{g}_6$ and $\hat{g}_{11}$, with coefficients of proportionality -4 and -2 , respectively. Importantly, we just need to fill the terms above (or below) the diagonal of the structure matrix. Symdyn automatically calculates the missing terms using the antisymmetry of the commutator. It is worth noting that we can incorporate symbolic entries in γ . As an example, for the low-order algebras of the previous section, using Table I(a) we can code γ including ϵ and υ as

```
import sympy
epsilon = sympy.Symbol('\epsilon')
upsilon = sympy.Symbol('\upsilon')
```

```
gamma[0, 1, 0] = -upsilon
gamma[0, 2, 1] = -2*epsilon
gamma[1, 2, 2] = -upsilon
```

Once γ has been introduced, we can initialize the Algebra class as

```
algebra = Algebra(gamma)
```

To continue benchmarking the library, a single similarity transformation, for instance $e^{\Lambda_2 \hat{g}_2} \hat{g}_4 e^{-\Lambda_2 \hat{g}_2}$, can be calculated using the code

```
algebra.get_similarity_transform(2, 4)
```

from which `sympdyn` will return a tuple with both an expression in $\mathcal{L}\mathcal{T}\mathcal{E}\mathcal{X}$ and the resulting Lie vector. In this case we get

$$-2\Lambda_2 \hat{g}_3 + \hat{g}_4, \quad (62)$$

as expected. Other complex calculations, as nested similarity transformations of Lie vectors

$$(e^{\Lambda_k \text{ad}_{\hat{g}_k}})(e^{\Lambda_j \text{ad}_{\hat{g}_j}}) \dots (e^{\Lambda_i \text{ad}_{\hat{g}_i}}) \sum_{l=1}^L \chi_l \hat{g}_l, \quad (63)$$

where $i, j, k \in \{1, 2, \dots, L\}$, can also be calculated with the library [97]. See our GitLab repository for further details on the usage of this routine, available under `algebra.get_nested_similarity_transform`.

The next key result of the WNM are the BCH matrices, which are obtained using the code

```
b_matrices = algebra.b_matrices
```

displaying all possible similarity transformations between the generators. Due to the size of these matrices, we prefer not to show them explicitly here, although the reader can find them in our repository. The next key result of the WNM is the coupling matrix ξ , which is obtained using the code

```
algebra.xi_matrix
```

Although we will not explicitly show this matrix here for the same reason than for the BCH matrices, we calculated it together with the coupled equations resulting from Eq. (35) in our library, which are obtained using the code

```
algebra.get_coupled_differential_equations()
```

These coupled equations agree with those in Ref. [23], so we validate our numerical implementation for high orders.

We can go further and calculate the determinant of the coupling matrix to find out if the solution is global. This is done with the code

```
sympy.det(algebra.xi_matrix)
```

For our current system we obtain $\det|\xi| = \exp(-3(\Lambda_6 + \Lambda_7))$, which verifies to be nonsingular for

all times except in the limit $(\Lambda_6 + \Lambda_7) \rightarrow \infty$. Accordingly, we can continue with the WNM in `sympdyn` to obtain the Wei-Norman equations [Eq. (36)]. This is done using the code

```
algebra.get_decoupled_differential_equations()
```

which, for our current example, returns

$$i\dot{\Lambda}_1 = 4\Lambda_1^2 \eta_8 + 2\Lambda_1 \Lambda_3 \eta_{10} + 2\Lambda_1 \eta_6 + \Lambda_3^2 \eta_9 + \Lambda_3 \eta_4 + \eta_1, \quad (64)$$

$$i\dot{\Lambda}_2 = 4\Lambda_2^2 \eta_9 + 2\Lambda_2 \Lambda_3 \eta_{10} + 2\Lambda_2 \eta_7 + \Lambda_3^2 \eta_8 + \Lambda_3 \eta_5 + \eta_2, \quad (65)$$

$$i\dot{\Lambda}_3 = \Lambda_3^2 \eta_{10} + 4\Lambda_3 (\Lambda_1 \eta_8 + \Lambda_2 \eta_9) + \Lambda_3 (\eta_6 + \eta_7) + 2\Lambda_2 (2\Lambda_1 \eta_{10} + \eta_4) + 2\Lambda_1 \eta_5 + \eta_3, \quad (66)$$

$$i\dot{\Lambda}_4 = -\Lambda_4^2 (2\Lambda_3 \eta_8 + 2\Lambda_2 \eta_{10} + \eta_5) - 4\Lambda_4 (\Lambda_2 \eta_9 - \Lambda_1 \eta_8) + \Lambda_4 (\eta_6 - \eta_7) + 2\Lambda_3 \eta_9 + 2\Lambda_1 \eta_{10} + \eta_4, \quad (67)$$

$$i\dot{\Lambda}_5 = 2\Lambda_4 \Lambda_5 (2\Lambda_3 \eta_8 + 2\Lambda_2 \eta_{10} + \eta_5) + 4\Lambda_5 (\Lambda_2 \eta_9 - \Lambda_1 \eta_8) + \Lambda_5 (\eta_7 - \eta_6) + 2\Lambda_3 \eta_8 + 2\Lambda_2 \eta_{10} + \eta_5, \quad (68)$$

$$i\dot{\Lambda}_6 = -\Lambda_4 (2\Lambda_3 \eta_8 + 2\Lambda_2 \eta_{10} + \eta_5) + 4\Lambda_1 \eta_8 + \Lambda_3 \eta_{10} + \eta_6, \quad (69)$$

$$i\dot{\Lambda}_7 = \Lambda_4 (2\Lambda_3 \eta_8 + 2\Lambda_2 \eta_{10} + \eta_5) + 4\Lambda_2 \eta_9 + \Lambda_3 \eta_{10} + \eta_7, \quad (70)$$

$$i\dot{\Lambda}_8 = \Lambda_5^2 (\Lambda_4^2 \eta_8 + \Lambda_4 \eta_{10} + \eta_9) e^{2\Lambda_6} + \Lambda_5 (2\Lambda_4 \eta_8 + \eta_{10}) e^{2\Lambda_6} + \eta_8 e^{2\Lambda_6}, \quad (71)$$

$$i\dot{\Lambda}_9 = (\Lambda_4^2 \eta_8 + \Lambda_4 \eta_{10} + \eta_9) e^{2\Lambda_7}, \quad (72)$$

$$i\dot{\Lambda}_{10} = 2\Lambda_5 (\Lambda_4^2 \eta_8 + \Lambda_4 \eta_{10} + \eta_9) e^{\Lambda_6 + \Lambda_7} + 2\Lambda_4 \eta_8 e^{\Lambda_6 + \Lambda_7} + \eta_{10} e^{\Lambda_6 + \Lambda_7}, \quad (73)$$

$$i\dot{\Lambda}_{11} = 2\Lambda_1 \eta_8 + 2\Lambda_2 \eta_9 + \Lambda_3 \eta_{10} + \eta_{11}. \quad (74)$$

Note that the first three equations form a block of coupled Riccati equations, which once solved, allows solving the fourth equation, which is also a Riccati equation. Finally, the rest can be solved by quadrature. As we previously mentioned, this hierarchy of Riccati equations is expected, provided the basis of the Lie algebra is a CWB [6,7]. Also note that the determinant of the coupling matrix is given by the exponential of a linear combination of coefficients Λ_6 and Λ_7 , being the ones related with the generators of the Cartan subalgebra. To the authors' best knowledge, this is the first time that the above equations have been calculated in its final form.

Finally, as mentioned before, changing the order for the exponential generators in the TEO involves recalculating γ . For instance, let us consider the new configuration for the TEO of our current example

$$\hat{U} = e^{\Lambda_1 \hat{g}_5} e^{\Lambda_2 \hat{g}_1} e^{\Lambda_3 \hat{g}_2} e^{\Lambda_4 \hat{g}_7} e^{\Lambda_5 \hat{g}_{11}} e^{\Lambda_6 \hat{g}_3} e^{\Lambda_7 \hat{g}_4} e^{\Lambda_8 \hat{g}_9} e^{\Lambda_9 \hat{g}_6} e^{\Lambda_{10} \hat{g}_8} e^{\Lambda_{11} \hat{g}_{10}}. \quad (75)$$

To calculate the γ it must be introduced the code

```
new_ord = (5, 1, 2, 7, 11, 3, 4, 9, 6, 8, 10)
```

```
new_alg = algebra.change_generators_order(new_ord)
```

This code swaps the rows, columns and layers of the initial γ accordingly, and thus a new Algebra object with the new structure tensor is created. After this, the user can obtain key results of the WNM from the new γ in the same way we did before.

We end this section by remarking that, once the Cartan-Weyl decomposition yields exponentiation of finite-nilpotent or diagonal matrices, the derivation of the Wei-Norman equations—as well as all intermediate steps—are carried out symbolically in sympy. In the next section, we specialize our library to the Lie group $SU(N)$ and present an example illustrating the direct introduction of finite matrix generators into the library to configure the structure tensor of the system.

IV. $SU(N)$

Without a doubt, $SU(N)$ is one of the most important groups in physics, playing a special role in quantum mechanics, quantum field theory, and the standard model of particle physics [98]. In particular, $SU(2)$ and $SU(3)$ describe key aspects of particle interactions as isospin symmetry [1] and the strong force among quarks and gluons [99], respectively. Larger groups like $SU(5)$ have been used to extend these symmetries in attempts to unify all fundamental interactions in grand unified theories [100].

In quantum computation, the role of $SU(N)$ is foundational, as a mathematical framework for describing quantum gates and transformations, and providing tools for error correction and algorithms optimization [63,101,102]. For qubit-based quantum computing, the special case where the group dimension is a power of two ($N = 2^n$) is considered, with n indicating the number of qubits in the system. Indeed, and as we showed in Sec. II C, the case $n = 1$ is the usual algebra to describe a time-dependent qubit. On the other hand, $SU(3)$ has also been used to describe quantum computing but with qutrits, which allow more efficient encoding of information and computations [103–105]. Higher-order computing with qudits, described by $SU(d)$, is also possible [106].

In this section, we present a generic CWB for $\mathfrak{su}(N)$ and show how it can be leveraged in symdyn using $\mathfrak{su}(2)$ as an example (Sec. IV A). We then use the library to obtain some key results of the method for $SU(3)$ and $SU(4)$, including the TEO in its explicit matrix form. We use $SU(3)$ to exemplify the introduction of finite matrix generators into the library (Sec. IV B), and finish by computing the CNOT gate, completing the gates needed for universal quantum computation (Sec. IV C).

A. Generic CWB for $\mathfrak{su}(N)$

Here, we define a generic CWB for $\mathfrak{su}(N)$, allowing us to apply the WNM in symdyn for an arbitrary N . Recall that this algebra comprises $L = N^2 - 1$ generators, which are $N \times N$ traceless matrices. To establish a CWB, we start by constructing a Cartan subalgebra $\mathfrak{h}$, which is composed of the maximum number of diagonalizable $N \times N$ matrices, forming a maximal commutative (Abelian) algebra [6,7]. By the traceless condition, the number of generators in $\mathfrak{h}$ is $N - 1$, defined as the rank of the algebra. Let us identify these generators

by $\hat{h}_j$, so that $j = 1, 2, \dots, N - 1$. One simple option, is to define them as diagonal matrices with two nonzero entries forming blocks of 1 and -1 , namely,

$$\hat{h}_j \equiv \text{diag}(0, \dots, 0, \underbrace{1}_j, \underbrace{-1}_{j+1}, 0, \dots, 0). \quad (76)$$

On the other hand, the generators of the root spaces correspond to the off-diagonal $N \times N$ matrices $\hat{r}_{nk}$, with all zero elements except a single 1 in entry (n, k) , representing raising ($n < k$) and lowering ($n > k$) operators. The number of generators of each root space is given by half the difference between the algebraic order and its rank, namely, $\frac{N(N-1)}{2}$. Finally, we choose the ordering of the generators to be the following:

$$\hat{g}_l = \begin{cases} \hat{r}_{nk}, & \text{for } l = f_1(n, k), \text{ with } n < k, \\ \hat{h}_j, & \text{for } l = \frac{N(N-1)}{2} + j, \\ \hat{r}_{nk}, & \text{for } l = \frac{N^2+N-2}{2} + f_2(n, k), \text{ with } n > k, \end{cases} \quad (77)$$

where $f_1(n, k) = k + N(n - 1) - \frac{n}{2}(n + 1)$ and $f_2(n, k) = k + \frac{1}{2}(n - 1)(n - 2)$, with $n, k = 1, 2, \dots, N$. Notice that we generically ordered the basis such that positive roots generators are first, second are the Cartan subalgebra generators, and the negative roots generators are third.

The above results provide a way to systematically construct this CWB for any N of $\mathfrak{su}(N)$. In symdyn, these generators can be built as follows:

```
from symdyn.sun.utils import get_sun_generators
generators = get_sun_generators(N)
```

with N specified by the user. Then, the structure tensor of the system is built as

```
from symdyn.utils import get_gamma
gamma = get_gamma(generators)
```

and the TEO in its explicit form can be obtained with the code

```
from symdyn import get_TEO
get_TEO(generators)
```

As an example, but also for the sake of completeness, let us obtain from symdyn the generators and the Wei-Norman equations for $\mathfrak{su}(2)$ with the above defined CWB. First, we use the code

```
algebra.get_sun_generators(2)
```

to load the basis. The generators can be printed, and they are

$$\hat{g}_1 = \begin{bmatrix} 0 & 1 \\ 0 & 0 \end{bmatrix}, \quad \hat{g}_2 = \begin{bmatrix} 1 & 0 \\ 0 & -1 \end{bmatrix}, \quad \hat{g}_3 = \begin{bmatrix} 0 & 0 \\ 1 & 0 \end{bmatrix}. \quad (78)$$

Using this basis, it can be built the structure tensor, and from it the coupling matrix, which determinant is $\det|\xi| =$

$\exp(-2\Lambda_2)$, and consequently we can obtain the Wei-Norman equations

$$i\dot{\Lambda}_1 = -\eta_3\Lambda_1^2 + 2\eta_2\Lambda_1 + \eta_1, \quad (79)$$

$$i\dot{\Lambda}_2 = -\eta_3\Lambda_1 + \eta_2, \quad (80)$$

$$i\dot{\Lambda}_3 = \eta_3e^{2\Lambda_2}. \quad (81)$$

The first equation is a Riccati equation, which once solved, allows solving the rest by quadrature. This is the expected structure of the differential equations provided the basis of the Lie algebra is a CWB. Note that the determinant of the coupling matrix only depends on the generator of the Cartan subalgebra. Finally, the explicit TEO is given by

$$\hat{U} = e^{-\Lambda_2} \begin{bmatrix} e^{2\Lambda_2} + \Lambda_1\Lambda_3 & \Lambda_1 \\ \Lambda_3 & 1 \end{bmatrix}. \quad (82)$$

$$\begin{aligned} \hat{g}_1 &= \begin{bmatrix} 0 & 1 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \hat{g}_2 = \begin{bmatrix} 0 & -i & 0 \\ i & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \hat{g}_3 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \hat{g}_4 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \hat{g}_5 = \begin{bmatrix} 0 & 0 & -i \\ 0 & 0 & 0 \\ i & 0 & 0 \end{bmatrix}, \\ \hat{g}_6 &= \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 1 & 0 \end{bmatrix}, \quad \hat{g}_7 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & -i \\ 0 & i & 0 \end{bmatrix}, \quad \hat{g}_8 = \frac{1}{\sqrt{3}} \begin{bmatrix} 1 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -2 \end{bmatrix}, \end{aligned} \quad (83)$$

satisfying the commutators

$$[\hat{g}_j, \hat{g}_k] = 2i \sum_{l=1}^8 f_{jkl} \hat{g}_l, \quad (84)$$

with $\gamma_{jkl} = 2if_{jkl}$ the structure constants. Of the latter, the only ones that have a value different from zero are

$$f_{123} = 1, \quad f_{458} = f_{678} = \frac{\sqrt{3}}{2}, \quad (85)$$

$$f_{147} = f_{165} = f_{246} = f_{257} = f_{345} = f_{376} = \frac{1}{2}, \quad (86)$$

being completely antisymmetric in all three indices. Accordingly, the Hamiltonian can be written as

$$\hat{H}(t) = \sum_{l=1}^8 \varsigma_l(t) \hat{g}_l, \quad (87)$$

and the correspondent TEO as

$$\hat{U}(t) = \prod_{l=1}^8 e^{\Theta_l(t) \hat{g}_l}. \quad (88)$$

To introduce the generators in Eqs. (83) in the library, L zero-filled matrices of dimension N are created with the code

Note that the basis in Eq. (78) differs from that in Table I for $\mathfrak{su}(2)$ only by a constant in the generator of the Cartan subalgebra $\hat{g}_2$. Consequently, the determinant of the coupling matrices and the explicit TEOs are slightly different in these bases.

B. $\mathfrak{su}(3)$

As mentioned before, $\mathfrak{su}(3)$ is a relevant algebra to describe many important physical systems. Further, it is usual to use the Gell-Mann matrices as its basis. However, by its cyclic structure (this basis can be considered as a generalization of the Pauli basis), we will need to change it to a CWB to get a factorized version of the correspondent TEO.

The Gell-Mann matrices are as follows:

```
generators = np.zeros((L, N, N))
```

In our case, L= 8 and N= 3. Then, the entries that are different from zero in each generator must be informed. For instance, generators $\hat{g}_1$ and $\hat{g}_8$ in Eq. (83) are introduced as

```
# First generator:
```

```
generators[0, 0, 1] = 1
```

```
generators[0, 1, 0] = 1
```

```
# Eight generator:
```

```
generators[7, 0, 0] = 1 / np.sqrt(3)
```

```
generators[7, 1, 1] = 1 / np.sqrt(3)
```

```
generators[7, 2, 2] = -2 / np.sqrt(3)
```

Once all the generators has been introduced, `sympdyn` automatically calculates their commutators [107], and with them the correspondent structure tensor. When working with higher-order Lie algebras with generators satisfying cyclic commutators, such as Gell-Mann matrices, this property can be used to automate the construction of the basis, as we show in our repository.

We can proceed with calculating the system's coupling matrix and its determinant. Similar to the Pauli basis for $\mathfrak{su}(2)$, the Gell-Mann basis produces a coupling matrix with periodic singularities that prevents the solution to be global. Owing to the size of this matrix, we have chosen not to display it explicitly here; however, it is available in our repository. As a consequence, a factorized version of the TEO cannot be obtained using this basis. Nevertheless, using a CWB the situation changes. For example, in one of the pioneering applications of the WNM for this algebra [108], the authors considered the group $SU(2) \times SU(2) \times SU(2)$ instead of $SU(3)$ directly, once the latter can be obtained from the former through a transformation, and used a CWB for the underlying

algebra of each $SU(2)$ subgroup, effectively simplifying the problem.

$$\begin{aligned} \hat{g}_1 &= \begin{bmatrix} 0 & 1 & 0 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \hat{g}_2 = \begin{bmatrix} 0 & 0 & 1 \\ 0 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \hat{g}_3 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 1 \\ 0 & 0 & 0 \end{bmatrix}, \quad \hat{g}_4 = \begin{bmatrix} 1 & 0 & 0 \\ 0 & -1 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \hat{g}_5 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 1 & 0 \\ 0 & 0 & -1 \end{bmatrix}, \\ \hat{g}_6 &= \begin{bmatrix} 0 & 0 & 0 \\ 1 & 0 & 0 \\ 0 & 0 & 0 \end{bmatrix}, \quad \hat{g}_7 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 1 & 0 & 0 \end{bmatrix}, \quad \hat{g}_8 = \begin{bmatrix} 0 & 0 & 0 \\ 0 & 0 & 0 \\ 0 & 1 & 0 \end{bmatrix}, \end{aligned} \quad (89)$$

which can be loaded in `symdyn` with `generators = get_sun_generators(3)`. The TEO now reads

$$\hat{U}(t) = \prod_{l=1}^8 e^{\Lambda_l(t)\hat{g}_l}. \quad (90)$$

We can continue to ask `symdyn` for the coupling matrix of the system (not shown here) and its determinant, the latter resulting in $\det|\xi| = \exp(-2(\Lambda_4 + \Lambda_5))$, involving the coefficients of the generators belonging to the Cartan subalgebra. Accordingly, we can ask `symdyn` for the Wei-Norman equations, which returns

$$i\dot{\Lambda}_1 = -\Lambda_1^2\eta_6 - \Lambda_1\Lambda_2\eta_7 + \Lambda_1(2\eta_4 - \eta_5) - \Lambda_2\eta_8 + \eta_1, \quad (91)$$

$$i\dot{\Lambda}_2 = -\Lambda_2^2\eta_7 - \Lambda_1\Lambda_2\eta_6 + \Lambda_2(\eta_4 + \eta_5) - \Lambda_1\eta_3 + \eta_2, \quad (92)$$

$$i\dot{\Lambda}_3 = -\Lambda_3^2(\eta_8 - \Lambda_1\eta_7) + \Lambda_3(\Lambda_1\eta_6 - \Lambda_2\eta_7) + \Lambda_3(2\eta_5 - \eta_4) + \Lambda_2\eta_6 + \eta_3, \quad (93)$$

$$i\dot{\Lambda}_4 = -\Lambda_1\eta_6 - \Lambda_2\eta_7 + \eta_4, \quad (94)$$

$$i\dot{\Lambda}_5 = -\Lambda_1\Lambda_3\eta_7 - \Lambda_2\eta_7 - \Lambda_3\eta_8 + \eta_5, \quad (95)$$

$$i\dot{\Lambda}_6 = (\eta_6 - \Lambda_3\eta_7)e^{(2\Lambda_4 - \Lambda_5)}, \quad (96)$$

$$i\dot{\Lambda}_7 = \Lambda_6(\Lambda_1\eta_7 + \eta_8)e^{(-\Lambda_4 + 2\Lambda_5)} + \eta_7e^{(\Lambda_4 + \Lambda_5)}, \quad (97)$$

$$i\dot{\Lambda}_8 = (\Lambda_1\eta_7 + \eta_8)e^{(-\Lambda_4 + 2\Lambda_5)}. \quad (98)$$

Note that the first two equations (the same number that the rank of this algebra) form a block of coupled Riccati equations, which once solved, allows solving the third equation, which is also a Riccati equation. Then, the rest can be solved by quadrature. As mentioned, this hierarchy of Riccati equations is expected. Finally, the explicit TEO is obtained

$$\hat{U} = e^{-\Lambda_5} \times \begin{bmatrix} \Lambda_1\Lambda_6\varrho_1 + \Lambda_7\varrho_2 + e^{\Lambda_4\Lambda_5} & \Lambda_1\varrho_1 + \Lambda_8\varrho_2 & \varrho_2 \\ \Lambda_3\Lambda_7 + \Lambda_6\varrho_1 & \Lambda_3\Lambda_8 + \varrho_1 & \Lambda_3 \\ \Lambda_7 & \Lambda_8 & 1 \end{bmatrix}, \quad (99)$$

where $\varrho_1 = e^{-\Lambda_4 + 2\Lambda_5}$ and $\varrho_2 = \Lambda_1\Lambda_3 + \Lambda_2$. To the authors' best knowledge, this is the first time that the above equations and explicit TEO for this algebra have been calculated using the WNM directly.

Taking $N = 3$ in Eq. (77), we obtain the following CWB for $\mathfrak{su}(3)$:

C. $\mathfrak{su}(4)$ and beyond

Here, we concentrate in the derivation of the CNOT gate from the explicit TEO of $\mathfrak{su}(4)$ given in Eq. (E18), and to discuss the behavior of `symdyn` for high-order quantum systems. The Wei-Norman equations and other key results of the WNM for this algebra can be found in Appendix E.

The CNOT gate is given by the 4×4 matrix

$$\hat{U}_{\text{CNOT}} = \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \quad (100)$$

In Appendix E, we determine the constraints that the TEO coefficients $\{\Lambda_i\}_{i=1}^{15}$ must meet for the TEO of Eq. (E18) works as a CNOT gate. There, we introduce a global phase ϕ_n to work directly with the TEO coefficients without applying the unitary condition. This phase is shown to be $\phi_n = \frac{(2n+1)\pi}{4}$ with $n \in \mathbb{Z}$. At certain application time, the TEO coefficients must satisfy the constraints $\Lambda_6 = \Lambda_{15} = e^{\Lambda_9 + i\phi_n}$, $\Lambda_7 = i\phi_n$, $\Lambda_8 = i\pi(\frac{1}{2} - n)$, $\Lambda_{11} = \Lambda_{13}e^{(-i\phi_n - \Lambda_9)}$, $\Lambda_{12} = \Lambda_{14}e^{(-i\phi_n - \Lambda_9)}$, $\Lambda_1 = \Lambda_2 = \Lambda_3 = \Lambda_4 = \Lambda_5 = \Lambda_{10} = 0$, Λ_{13} and Λ_{14} are free complex parameters, and $\Lambda_9 \rightarrow \infty$. In such limit, the TEO matches the CNOT gate up to the global phase ϕ_n . We anticipate that, due to the complexity of the Wei-Norman equations for $\mathfrak{su}(4)$, Eqs. (E3)–(E17), a more in-depth study for designing a Hamiltonian that realizes the CNOT gate will be needed (see, for instance, Ref. [109]). This and other kind of problems will be subject of future work.

The CNOT gate, together with the H and T gates obtained in Sec. II C, completes a set of gates for universal computation. Their derivation from explicit TEOs demonstrates that applying the WNM to $\mathfrak{su}(N)$ endorsed with a CWB, handles matrix exponentiation very well.

We conclude this section by examining the scaling behavior and practical limitations of the WNM. For this purpose, we measure the computation time τ required by `symdyn` to generate the Wei-Norman equations for $\mathfrak{su}(N)$ on a standard laptop [110], as N increases. Recall that the number of generators grows with the group (or algebra) dimension as $L = N^2 - 1$, and that this algebra describes a system of n -coupled qubits whenever $N = 2^n$. Figure 3 shows the measured values of τ as a function of N , along with a fitted curve, revealing that the computation time grows exponentially with the algebra dimension. Finally, we recall that the maximum number of coupled Riccati equations appearing in the Wei-Norman

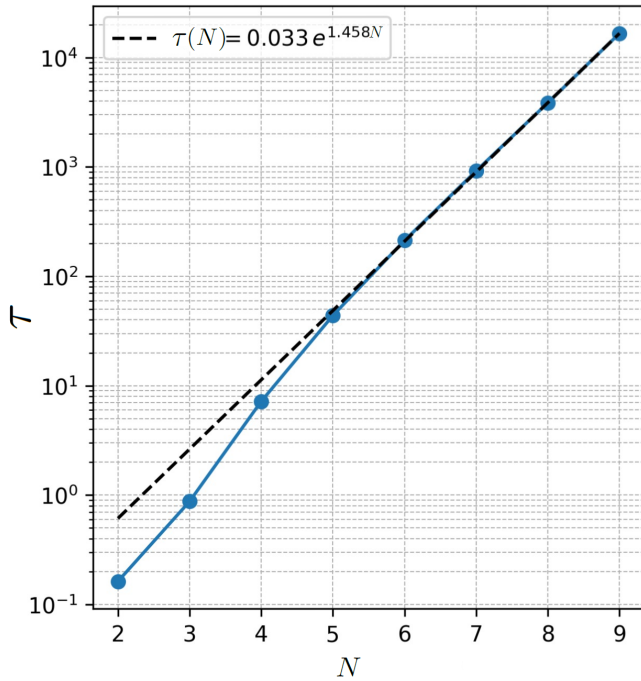


FIG. 3. Computation time of the Wei-Norman equations in Symdyn as a function of the dimension of $\mathfrak{su}(N)$. The fit of $\tau(N)$ only includes the last five points.

system is $N - 1$. Assessing the complexity and numerical stability of the solution of these equations is left for future work.

V. CONCLUSIONS AND PERSPECTIVES

In this work, we have introduced *symdyn*, a Python library designed to automate the WNM, overcoming challenges associated with high-dimensional quantum systems. Our library facilitates the computation of the factorized TEO, regardless of the ordering of exponentials in the factorization, and enables the calculation of simple and nested commutators, as well as similarity transformations. We have emphasized the crucial role of selecting an appropriate basis to maximize the method's utility and have analytically recovered nontrivial results for low-order QLSs as examples.

Furthermore, we have demonstrated *symdyn*'s power and flexibility in the treatment of high-order QLSs by deriving the Wei-Norman equations for a system of two coupled time-dependent harmonic oscillators with time-dependent coupling. We have specialized our library for the specially unitary group $SU(N)$ by providing a generic CWB for its associated algebra, making it applicable for arbitrary N . We applied this framework to $SU(2)$, $SU(3)$, and $SU(4)$, and used our results to obtain the Hadamard, T , and CNOT gates, completing the necessary gates for universal quantum computation.

There is a growing interest in understanding the interplay between increasing the dimension of Lie algebras, linked to the number of components in a quantum system, and the ability to compute their time evolution [17,102,111]. We expect our results for $SU(N)$ to contribute to tackling these scaling challenges. Moreover, we expect *symdyn* to become a power-

ful tool for studying high-dimensional quantum systems and exploring frontiers in quantum dynamics, optimal control, and quantum computation.

Currently, we are developing a method which leverages the composition property of Lie groups to provide explicit solutions for the TEO of QLSs. This extension aims to enhance the efficient resolution of the Wei-Norman equations, also addressing their stability [112,113]. Additionally, we plan to expand our repository by incorporating QLSs and analyzing their respective differential equations.

In the near future, we aim to extend the WNM to the algebras of pseudo-orthogonal groups, which play a fundamental role in high-energy physics and interacting dynamical systems. These groups offer a symmetry-based approach to understanding quantum and classical correlations. In particular, the Lorentz group has widespread applications in high-energy physics, where quantum computation frameworks are being actively explored [114].

ACKNOWLEDGMENTS

The authors thank J. A. Helayël Neto, J. I. Rubiano-Murcia, and A. C. Duriez for their valuable and enriching discussions. D.M.T. also expresses gratitude to Sebastian Galliez Martínez and L. Pires for their assistance with the graphical representation of the structure tensor. D.M.T. and A.Z.K. also acknowledge the support of Brazilian scientific and technological research agencies, including the Coordination for the Improvement of Higher Education Personnel (CAPES), the National Council for Scientific and Technological Development (CNPq), the Fundação de Amparo à Pesquisa do Estado de São Paulo (FAPESP), Grant No. 2021/06823-5 (A.Z.K.) and the Fundação de Amparo à Pesquisa do Estado do Rio de Janeiro (FAPERJ), Grants No. E-26/204.433/2021 (D.M.T.) and No. E-26/203.939/2024 (A.Z.K.), for partial financial support.

DATA AVAILABILITY

The data that support the findings of this article are openly available in our GitLab repository at [65].

APPENDIX A: WEI-NORMAN FACTORIZATION

Here, we show that an arbitrary element of a Lie group given in the unfactorized representation of Eq. (4), can equivalently be expressed in the factorized representation of Eq. (5) if ξ is invertible for all times. Recall that the functions relating the Λ_l and λ_l coefficients are known as BCH-like relations [25], and that we are considering the ordering given in Eq. (7). Following standard algebraic techniques, we redefine the operators in Eqs. (4) and (5), as the special case $\theta = 1$ of

$$\hat{F}(\theta) = \exp\left(\sum_{l=1}^L \theta \lambda_l \hat{g}_l\right) \quad (\text{A1})$$

and

$$\hat{F}(\theta) = \prod_{l=1}^L e^{\Lambda_l(\theta) \hat{g}_l}, \quad (\text{A2})$$

respectively. Notice that we have suppressed the explicit time-dependence in the coefficients of the above equations once

our results relating both representations are valid for the same arbitrary instant of time. The idea is to differentiate the above representations of $\hat{F}$ with respect to θ , rewrite them proportional to $\hat{F}$, and then impose their derivatives to be equal, establishing the set of differential equations that relates the set of coefficients in both representations.

The derivative of Eq. (A1) is direct, and given by

$$\hat{F}' = \sum_{l=1}^L \lambda_l \hat{g}_l \hat{F}, \quad (\text{A3})$$

where the prime indicates derivative with respect to θ . On the other hand, the derivative of a factorized element with respect to any parameter is equivalent to the calculated in Eq. (30). Therefore, for Eq. (A2) we have

$$\hat{F}' = \sum_{l=1}^L \sum_{n=1}^L \Lambda_n' \xi_{nl} \hat{g}_l \hat{F}. \quad (\text{A4})$$

Using the linear independence of the generators, equating Eqs. (A3) and (A4) is obtained

$$\lambda_l = \sum_{n=1}^L \Lambda_n' \xi_{nl}, \quad (\text{A5})$$

which is a set of coupled differential equations with the initial condition $\Lambda_n = 0$ for $\theta = 0$, and can be written in matrix notation as

$$\underbrace{\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \vdots \\ \lambda_L \end{bmatrix}}_{\boldsymbol{\lambda}} = \underbrace{\begin{bmatrix} \xi_{11} & \xi_{12} & \dots & \xi_{1L} \\ \xi_{21} & \xi_{22} & \dots & \xi_{2L} \\ \vdots & \vdots & \ddots & \vdots \\ \xi_{L1} & \xi_{L2} & \dots & \xi_{LL} \end{bmatrix}}_{\boldsymbol{\xi}^T} \underbrace{\begin{bmatrix} \Lambda_1' \\ \Lambda_2' \\ \vdots \\ \Lambda_L' \end{bmatrix}}_{\boldsymbol{\Lambda}'}. \quad (\text{A6})$$

The above system of differential equations is almost identical to such in Eq. (35) with the following differences: first and more important, there is not explicit dependence of λ_l on the parameter θ . Second, there is no imaginary unit at the right-hand side of the equation. Accordingly, for a coupling matrix that is all time invertible we have

$$(\boldsymbol{\xi}^T)^{-1} \boldsymbol{\lambda} = \boldsymbol{\Lambda}', \quad (\text{A7})$$

which are the Wei-Norman equations for the factorization procedure. Recall that once the system is solved, then it must be evaluated at $\theta = 1$ to obtain the desired BCH-like relations.

To illustrate the above results, we next calculate the BCH-like relations for $\mathfrak{su}(1, 1)$, $\mathfrak{su}(2)$, $\mathfrak{sl}(2)$ and $\mathfrak{so}(2, 1)$ QLSs with the commutators given in Table I(a). Using Eqs. (32) and (A6), we get

$$\begin{bmatrix} \lambda_1 \\ \lambda_2 \\ \lambda_3 \end{bmatrix} = \begin{bmatrix} 1 & -v\Lambda_1 & \epsilon v \frac{(\Lambda_1)^2}{e^{v\Lambda_2}} \\ 0 & 1 & -2\epsilon \frac{\Lambda_1}{e^{v\Lambda_2}} \\ 0 & 0 & \frac{1}{e^{v\Lambda_2}} \end{bmatrix} \begin{bmatrix} \Lambda_1' \\ \Lambda_2' \\ \Lambda_3' \end{bmatrix}. \quad (\text{A8})$$

As mentioned in Sec. II B, once $\det |\boldsymbol{\xi}| = \exp(-v\Lambda_2)$ one can invert $\boldsymbol{\xi}^T$ for all times, except for the limiting case $\Lambda_2 \rightarrow \infty$ which nonetheless is tractable, as shown in Appendix C. Then,

multiplying by $(\boldsymbol{\xi}^T)^{-1}$ from the left in the above equation, we obtain the Wei-Norman equations:

$$\Lambda_1' - \epsilon v \lambda_3 (\Lambda_1)^2 - v \lambda_2 \Lambda_1 - \lambda_1 = 0, \quad (\text{A9})$$

$$\Lambda_2' - 2\epsilon \lambda_3 \Lambda_1 - \lambda_2 = 0, \quad (\text{A10})$$

$$\Lambda_3' - \lambda_3 e^{v\Lambda_2} = 0. \quad (\text{A11})$$

The first equation above represents four families of parameter-independent Riccati equations, each associated with one of the Lie algebras we are considering. Once this Riccati equation is solved, the other two equations can be integrated by quadrature using the initial conditions. The analytical solution of Eq. (A9) is well-known (see, for instance, the Appendix of Ref. [115]). Moreover, the solution of the above equations is equivalent to the one in Refs. [22,116] for the Lie algebras $\mathfrak{su}(1, 1)$, $\mathfrak{su}(2)$, and $\mathfrak{so}(2, 1)$, but now we are including the Lie algebra $\mathfrak{sl}(2)$. This solution is given as

$$\Lambda_1 = \frac{2\lambda_1 \sinh(v)}{2v \cosh(v) - v\lambda_2 \sinh(v)}, \quad (\text{A12})$$

$$\Lambda_3 = \frac{2\lambda_3 \sinh(v)}{2v \cosh(v) - v\lambda_2 \sinh(v)}, \quad (\text{A13})$$

and

$$\Lambda_2 = -\frac{2}{v} \ln \left(\cosh(v) - \frac{v\lambda_2}{2v} \sinh(v) \right), \quad (\text{A14})$$

with

$$v^2 = \left(\frac{v\lambda_2}{2} \right)^2 - \epsilon v \lambda_1 \lambda_3. \quad (\text{A15})$$

The above equations are BCH-like relations for the Lie algebras $\mathfrak{su}(1, 1)$, $\mathfrak{su}(2)$, $\mathfrak{so}(2, 1)$, and $\mathfrak{sl}(2)$. Recall that a different choice for the order of the generators, which means reordering the elements of the structure tensor in Table I(a), will lead to a different set of functions for the above coefficients.

APPENDIX B: UNITARITY CONSTRAINTS FOR THE SU(2) TEO

Here, we deduce the constraints to guarantee the unitarity of the TEO in Eq. (53). From that equation, it follows directly that

$$\hat{U}^\dagger = e^{-\frac{\Lambda_2^*}{2}} \begin{bmatrix} e^{\Lambda_3^*} + \Lambda_1^* \Lambda_3^* & \Lambda_3^* \\ \Lambda_1^* & 1 \end{bmatrix}, \quad (\text{B1})$$

where $\dagger$ represents the Hermitian conjugate and $*$ the complex conjugate. The unitarity condition implies $\hat{U}^\dagger \hat{U} = \hat{U} \hat{U}^\dagger = \hat{1}$. Accordingly, using Eqs. (53) and (B1), it must be satisfied

$$\begin{bmatrix} 1 & 0 \\ 0 & 1 \end{bmatrix} = e^{-\Re(\Lambda_2)} \begin{bmatrix} |\kappa|^2 + |\Lambda_1|^2 & \kappa \Lambda_3^* + \Lambda_1^* \\ \kappa^* \Lambda_3 + \Lambda_1^* & 1 + |\Lambda_3|^2 \end{bmatrix}, \quad (\text{B2})$$

where $\kappa = e^{\Lambda_2} + \Lambda_1 \Lambda_3$, $\Re(z)$ and $\Im(z)$ represent the real and imaginary part of the complex number z , and the TEO coefficients are written in their polar form as $\Lambda_1 = |\Lambda_1| e^{i\varphi_1}$, $\Lambda_2 = |\Lambda_2| e^{i\varphi_2}$ and $\Lambda_3 = |\Lambda_3| e^{i\varphi_3}$. Once $e^{-\Re(\Lambda_2)} \neq 0$, then

Eq. (B2) implies the system of equations

$$e^{-\Re(\Lambda_2)}(|\kappa|^2 + |\Lambda_1|^2) = 1, \quad (\text{B3})$$

$$e^{-\Re(\Lambda_2)}(1 + |\Lambda_3|^2) = 1, \quad (\text{B4})$$

$$\kappa|\Lambda_3|e^{-i\varphi_3} + |\Lambda_1|e^{i\varphi_1} = 0. \quad (\text{B5})$$

Using Eq. (B5) and its complex conjugated we first calculate $|\kappa| = \frac{|\Lambda_1|}{|\Lambda_3|}$, and then substituting in Eq. (B3) we obtain

$$e^{-\Re(\Lambda_2)}(1 + |\Lambda_3|^2) = \frac{|\Lambda_3|^2}{|\Lambda_1|^2}. \quad (\text{B6})$$

From Eq. (B4) and the above equation it is clear that $|\Lambda_3| = |\Lambda_1|$, which is the first constraint. Equation (B4) can be now rewritten as $e^{\Re(\Lambda_2)} = (1 + |\Lambda_1|^2)$, which is the second constraint. Now, substituting κ explicitly in Eq. (B5) and using the first constraint we obtain

$$e^{\Lambda_2} + (1 + |\Lambda_1|^2)e^{i(\varphi_1 + \varphi_3)} = 0. \quad (\text{B7})$$

Finally, multiplying the above equation by $e^{-\Re(\Lambda_2)}$ and using the second constraint we get

$$e^{i\Im(\Lambda_2)} + e^{i(\varphi_1 + \varphi_3)} = 0, \quad (\text{B8})$$

which is the third constraint, and can be written as $\Im(\Lambda_2) = \varphi_1 + \varphi_3 + (2n + 1)\pi$, with $n \in \mathbb{Z}$. They are partially in agreement with those of Ref. [22] recalling that they considered Λ_2 as the coefficient of $\hat{g}_2$ and our third constraint coincides with Eq. (14) of that reference for $n = 1$.

APPENDIX C: X GATE IN THE CARTAN-WEYL BASIS

Here we consider a different approach than such in Sec. II C to obtain the X gate

$$\hat{U}_x = \begin{bmatrix} 0 & 1 \\ 1 & 0 \end{bmatrix}. \quad (\text{C1})$$

The idea is to start from Eq. (54), ask for the values that parameters $|\Lambda_1|$, φ_1 and φ_3 should take to get the mentioned gate, and obtain the Hamiltonian coefficients that produces them using the WNM. This is a kind of an inverse problem. Initially, to get a null diagonal, the only option is to have $|\Lambda_1| \rightarrow \infty$. Notice that this limit must also be taken as a step to obtain the Y gate, which is proportional to $\hat{\sigma}_2$. The

X gate is obtained, up to a global phase, if it is satisfied $\varphi_1 = \varphi_3 \equiv \varphi$, while to obtain the Y gate it must be satisfied $\varphi_1 = (2n + 3/2)\pi$, with $n \in \mathbb{Z}$, and $\varphi_3 = (2m + 1/2)\pi$, with $m \in \mathbb{Z}$. From the first two unitarity constraints we have that $|\Lambda_3| \rightarrow \infty$ and $e^{\Re(\Lambda_2)} \rightarrow \infty$, so it is evident that to obtaining such a gates it is necessary to take the limit $\Lambda_2 \rightarrow \infty$, in which the coupling matrix cannot be inverted (see Eq. (37)). Notice that the third unitarity constraint implies for the X gate that $e^{i\Im(\Lambda_2)} = -e^{i\varphi}$.

It is well known that the X gate can be obtained from a Hamiltonian proportional to $\hat{\sigma}_1$. Once our Hamiltonian in Eq. (55) is transformed to the CWB, the above implies to put $\eta_2 = 0$. We also define $\eta_1 = \eta_3 \equiv \eta$. This yields

$$\hat{H} = \eta(\hat{g}_1 + \hat{g}_3). \quad (\text{C2})$$

As the above Hamiltonian is time-independent, it follows directly that

$$\hat{U} = \exp(\lambda(\hat{g}_1 + \hat{g}_3)), \quad (\text{C3})$$

where $\lambda = i\eta t$. Accordingly, the problem has been reduced to a factorization problem, for which we can use Eqs. (A12)–(A15) with $\epsilon = -1$ and $v = 1$, $\lambda_2 = 0$ and $\lambda_1 = \lambda_3 = \lambda$, yielding $\Lambda_1 = \Lambda_3 = \tanh(\lambda)$ and $\Lambda_2 = -2 \ln \cosh(\lambda)$. In terms of the Hamiltonian coefficient, we obtain $\Lambda_1 = \Lambda_3 = -i \tan(\eta t)$ and $\Lambda_2 = -2 \ln \cos(\eta t)$. With these solutions, the TEO is explicitly

$$\hat{U}(t) = \begin{bmatrix} \cos(\eta t) & -i \sin(\eta t) \\ -i \sin(\eta t) & \cos(\eta t) \end{bmatrix}. \quad (\text{C4})$$

At $t = t^* := (n + 1/2)\pi/\eta$, with $n \in \mathbb{Z}$, the TEO $\hat{U}(t^*)$ is the X gate up to a global phase of $i^{(2n-1)}$.

APPENDIX D: TIME-DEPENDENT COUPLED OSCILLATORS

In this Appendix, we provide some details for the problem of two time-dependent harmonic oscillators coupled with a time-dependent coupling of Sec. III A. As we mentioned, the effective Hamiltonian of the system can be written as in Eq. (61), where the generators are bilinear products of ladder operators of each mode, and the coefficients are functions of the physical parameters of the system, i.e., the mirror position and its derivative, the frequency of the modes, and the cou-

TABLE III. Structure table for a system of two coupled time-dependent harmonic oscillators. Symbol $*$ is used here to indicate that such entries are known once this table is antisymmetric.

	$\hat{g}_1$	$\hat{g}_2$	$\hat{g}_3$	$\hat{g}_4$	$\hat{g}_5$	$\hat{g}_6$	$\hat{g}_7$	$\hat{g}_8$	$\hat{g}_9$	$\hat{g}_{10}$	$\hat{g}_{11}$
$\hat{g}_1$	0	0	0	0	$-2\hat{g}_3$	$-2\hat{g}_1$	0	$-4\hat{g}_6 - 2\hat{g}_{11}$	0	$-2\hat{g}_4$	0
$\hat{g}_2$	*	0	0	$-2\hat{g}_3$	0	0	$-2\hat{g}_2$	0	$-4\hat{g}_7 - 2\hat{g}_{11}$	$-2\hat{g}_5$	0
$\hat{g}_3$	*	*	0	$-\hat{g}_1$	$-\hat{g}_2$	$-\hat{g}_3$	$-\hat{g}_3$	$-2\hat{g}_5$	$-2\hat{g}_4$	$-\hat{g}_6 - \hat{g}_7 - \hat{g}_{11}$	0
$\hat{g}_4$	*	*	*	0	$\hat{g}_6 - \hat{g}_7$	$-\hat{g}_4$	$\hat{g}_4$	$-2\hat{g}_{10}$	0	$-\hat{g}_9$	0
$\hat{g}_5$	*	*	*	*	0	$\hat{g}_5$	$-\hat{g}_5$	0	$-2\hat{g}_{10}$	$-\hat{g}_8$	0
$\hat{g}_6$	*	*	*	*	*	0	0	$-2\hat{g}_8$	0	$-\hat{g}_{10}$	0
$\hat{g}_7$	*	*	*	*	*	*	0	0	$-2\hat{g}_9$	$-\hat{g}_{10}$	0
$\hat{g}_8$	*	*	*	*	*	*	*	0	0	0	0
$\hat{g}_9$	*	*	*	*	*	*	*	*	0	0	0
$\hat{g}_{10}$	*	*	*	*	*	*	*	*	*	0	0
$\hat{g}_{11}$	*	*	*	*	*	*	*	*	*	*	0

pling parameters being elements of a tensor representing all the intermode interactions [23]. Note that other types of interaction between modes can give rise to underlying nonclosed Lie algebras [117]. To construct the structure tensor, we shall write explicitly the generators of the system. Let $\hat{a}_j$ and $\hat{a}_j^\dagger$ be ladder operators of the mode j , with $j = 1, 2$. Accordingly, they satisfy

$$[\hat{a}_j, \hat{a}_k] = 0 \quad \text{and} \quad [\hat{a}_j, \hat{a}_k^\dagger] = \delta_{jk} \hat{1}. \quad (\text{D1})$$

The eleven generators are defined as

$$\begin{aligned} \hat{g}_1 &= \hat{a}_1^{\dagger 2}, & \hat{g}_2 &= \hat{a}_2^{\dagger 2}, & \hat{g}_3 &= \hat{a}_1^\dagger \hat{a}_2^\dagger, & \hat{g}_4 &= \hat{a}_1^\dagger \hat{a}_2, \\ \hat{g}_5 &= \hat{a}_1 \hat{a}_2^\dagger, & \hat{g}_6 &= \hat{a}_1^\dagger \hat{a}_1, & \hat{g}_7 &= \hat{a}_2^\dagger \hat{a}_2, & \hat{g}_8 &= \hat{a}_1^2, \\ \hat{g}_9 &= \hat{a}_2^2, & \hat{g}_{10} &= \hat{a}_1 \hat{a}_2, & \hat{g}_{11} &= \hat{1}. \end{aligned} \quad (\text{D2})$$

using Eqs. (D1), we calculate the commutators between the above generators and build the correspondent structure table of the system (Table III), displayed at the end of this Appendix. From it can be noted that $\hat{g}_6$ and $\hat{g}_7$ are members of the Cartan subalgebra, and with exception of $\hat{g}_{11}$, all other generators belong to the root spaces. Recall that the determinant of the correspondent coupling matrix involves the exponential of $\hat{g}_6$ and $\hat{g}_7$.

APPENDIX E: KEY RESULTS OF THE WNM FOR $\mathfrak{su}(4)$ AND DERIVATION OF THE CNOT GATE

Here, we use symdyn to calculate some key results of the WNM for $\mathfrak{su}(4)$. Initially, taking $N = 4$ in Eq. (77), we obtain the generic CWB for $\mathfrak{su}(3)$

$$\begin{aligned} \hat{g}_1 &= \begin{bmatrix} 0 & 1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \dots, & \hat{g}_6 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 1 \\ 0 & 0 & 0 & 0 \end{bmatrix}, & \hat{g}_7 &= \begin{bmatrix} 1 & 0 & 0 & 0 \\ 0 & -1 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \dots, \\ \hat{g}_9 &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \\ 0 & 0 & 0 & -1 \end{bmatrix}, & \hat{g}_{10} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 1 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \end{bmatrix}, \dots, & \hat{g}_{15} &= \begin{bmatrix} 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 0 & 0 \\ 0 & 0 & 1 & 0 \end{bmatrix}. \end{aligned} \quad (\text{E1})$$

The TEO now reads

$$\hat{U}(t) = \prod_{l=1}^{15} e^{\Lambda_l(t) \hat{g}_l}. \quad (\text{E2})$$

We can continue to ask symdyn for the determinant of the coupling matrix of the system, which results to be $\det|\xi| = \exp(-2(\Lambda_7 + \Lambda_8 + \Lambda_9))$, involving the coefficients of the generators belonging to the Cartan subalgebra. Accordingly, we can ask symdyn for the Wei-Norman equations, which returns

$$i\dot{\Lambda}_1 = -\Lambda_1^2 \eta_{10} - \Lambda_1(\Lambda_2 \eta_{11} + \Lambda_3 \eta_{12}) + \Lambda_1(2\eta_7 - \eta_8) - \Lambda_2 \eta_{13} - \Lambda_3 \eta_{14} + \eta_1, \quad (\text{E3})$$

$$i\dot{\Lambda}_2 = -\Lambda_2^2 \eta_{11} - \Lambda_2(\Lambda_1 \eta_{10} + \Lambda_3 \eta_{12}) + \Lambda_2(\eta_7 + \eta_8 - \eta_9) - \Lambda_3 \eta_{15} - \Lambda_1 \eta_4 + \eta_2, \quad (\text{E4})$$

$$i\dot{\Lambda}_3 = -\Lambda_3^2 \eta_{12} - \Lambda_3(\Lambda_1 \eta_{10} + \Lambda_2 \eta_{11}) + \Lambda_3(\eta_7 + \eta_9) - \Lambda_1 \eta_5 - \Lambda_2 \eta_6 + \eta_3, \quad (\text{E5})$$

$$i\dot{\Lambda}_4 = -\Lambda_4^2(\Lambda_1 \eta_{11} + \eta_{13}) - \Lambda_4 \Lambda_5(\Lambda_1 \eta_{12} + \eta_{14}) + \Lambda_4(\Lambda_1 \eta_{10} - \Lambda_2 \eta_{11} - \eta_7 + 2\eta_8 - \eta_9) - \Lambda_5(\Lambda_2 \eta_{12} + \eta_{15}) + \Lambda_2 \eta_{10} + \eta_4, \quad (\text{E6})$$

$$i\dot{\Lambda}_5 = -\Lambda_5^2(\Lambda_1 \eta_{12} + \eta_{14}) - \Lambda_4 \Lambda_5(\Lambda_1 \eta_{11} + \eta_{13}) + \Lambda_5(\Lambda_1 \eta_{10} - \Lambda_3 \eta_{12} - \eta_7 + \eta_8 + \eta_9) - \Lambda_4(\Lambda_3 \eta_{11} + \eta_6) + \Lambda_3 \eta_{10} + \eta_5, \quad (\text{E7})$$

$$i\dot{\Lambda}_6 = -\Lambda_6^2(\Lambda_2 \eta_{12} + \eta_{15} + \Lambda_4(\Lambda_1 \eta_{12} + \eta_{14})) - \Lambda_5 \Lambda_6(\Lambda_1 \eta_{12} + \eta_{14}) + \Lambda_6(\Lambda_2 \eta_{11} + \Lambda_4(\Lambda_1 \eta_{11} + \eta_{13}) - \Lambda_3 \eta_{12} - \eta_8 + 2\eta_9) + \Lambda_3 \eta_{11} + \Lambda_5(\Lambda_1 \eta_{11} + \eta_{13}) + \eta_6, \quad (\text{E8})$$

$$i\dot{\Lambda}_7 = -\Lambda_1 \eta_{10} - \Lambda_2 \eta_{11} - \Lambda_3 \eta_{12} + \eta_7, \quad (\text{E9})$$

$$i\dot{\Lambda}_8 = -\Lambda_2 \eta_{11} - \Lambda_3 \eta_{12} - \Lambda_4(\Lambda_1 \eta_{11} + \eta_{13}) - \Lambda_5(\Lambda_1 \eta_{12} + \eta_{14}) + \eta_8, \quad (\text{E10})$$

$$i\dot{\Lambda}_9 = -\Lambda_3 \eta_{12} - \Lambda_5(\Lambda_1 \eta_{12} + \eta_{14}) - \Lambda_6(\Lambda_2 \eta_{12} + \eta_{15} + \Lambda_4(\Lambda_1 \eta_{12} + \eta_{14})) + \eta_9, \quad (\text{E11})$$

$$i\dot{\Lambda}_{10} = (\eta_{10} - \Lambda_4 \eta_{11} - \Lambda_5 \eta_{12}) e^{(2\Lambda_7 - \Lambda_8)}, \quad (\text{E12})$$

$$i\dot{\Lambda}_{11} = \Lambda_{10}(\Lambda_1 \eta_{11} + \eta_{13} - \Lambda_6(\Lambda_1 \eta_{12} + \eta_{14})) e^{(-\Lambda_7 + 2\Lambda_8 - \Lambda_9)} + (\eta_{11} - \Lambda_6 \eta_{12}) e^{(\Lambda_7 + \Lambda_8 - \Lambda_9)}, \quad (\text{E13})$$

$$i\dot{\Lambda}_{12} = \Lambda_{10}(\Lambda_1 \eta_{12} + \eta_{14}) e^{(-\Lambda_7 + \Lambda_8 + \Lambda_9)} + \Lambda_{11}(\Lambda_2 \eta_{12} + \eta_{15} + \Lambda_4(\Lambda_1 \eta_{12} + \eta_{14})) e^{(-\Lambda_8 + 2\Lambda_9)} + \eta_{12} e^{(\Lambda_7 + \Lambda_9)}, \quad (\text{E14})$$

$$i\dot{\Lambda}_{13} = (\Lambda_1 \eta_{11} + \eta_{13} - \Lambda_6(\Lambda_1 \eta_{12} + \eta_{14})) e^{(-\Lambda_7 + 2\Lambda_8 - \Lambda_9)}, \quad (\text{E15})$$

$$i\dot{\Lambda}_{14} = \Lambda_{13}(\Lambda_2 \eta_{12} + \eta_{15} + \Lambda_4(\Lambda_1 \eta_{12} + \eta_{14})) e^{(-\Lambda_8 + 2\Lambda_9)} + (\Lambda_1 \eta_{12} + \eta_{14}) e^{(-\Lambda_7 + \Lambda_8 + \Lambda_9)}, \quad (\text{E16})$$

$$i\dot{\Lambda}_{15} = (\Lambda_2 \eta_{12} + \eta_{15} + \Lambda_4(\Lambda_1 \eta_{12} + \eta_{14})) e^{-\Lambda_8 + 2\Lambda_9}. \quad (\text{E17})$$

As can be seen, the first three equations (the same number as the rank of this algebra) form a block of coupled Riccati equations, which, once solved, allow us to solve the fourth and fifth equations, which are themselves coupled Riccati equations. Then, we can move on to solve the sixth equation, which is a Riccati equation, and the rest can be solved by quadrature. As mentioned, this hierarchy of Riccati equations is expected.

The explicit form of the TEO for this algebra is given as

$$\hat{U} = \begin{bmatrix} \Lambda_1 \Lambda_{10} \varrho_3 + \Lambda_{11} \varrho_4 + \Lambda_{13} \varrho_6 e^{-\Lambda_9} + e^{\Lambda_7} & \Lambda_{12} \varrho_4 + \Lambda_{14} \varrho_6 e^{-\Lambda_9} + \Lambda_1 \varrho_3 & \Lambda_{15} \varrho_6 e^{-\Lambda_9} + \varrho_4 & \varrho_6 e^{-\Lambda_9} \\ \Lambda_{10} \varrho_3 + \Lambda_{11} \Lambda_4 \varrho_5 + \Lambda_{13} \varrho_7 e^{-\Lambda_9} & \Lambda_{12} \Lambda_4 \varrho_5 + \Lambda_{14} \varrho_7 e^{-\Lambda_9} + \varrho_3 & \Lambda_{15} \varrho_7 e^{-\Lambda_9} + \Lambda_4 \varrho_5 & \varrho_7 e^{-\Lambda_9} \\ \Lambda_{11} \varrho_5 + \Lambda_{13} \Lambda_6 e^{-\Lambda_9} & \Lambda_{12} \varrho_5 + \Lambda_{14} \Lambda_6 e^{-\Lambda_9} & \Lambda_{15} \Lambda_6 e^{-\Lambda_9} + \varrho_5 & \Lambda_6 e^{-\Lambda_9} \\ \Lambda_{13} e^{-\Lambda_9} & \Lambda_{14} e^{-\Lambda_9} & \Lambda_{15} e^{-\Lambda_9} & e^{-\Lambda_9} \end{bmatrix}, \quad (\text{E18})$$

where $\varrho_3 = e^{(\Lambda_8 - \Lambda_7)}$, $\varrho_4 = (\Lambda_1 \Lambda_4 + \Lambda_2) \varrho_5$ with $\varrho_5 = e^{(-\Lambda_8 + \Lambda_9)}$, $\varrho_6 = \Lambda_1 \Lambda_5 + \Lambda_3 + \Lambda_6 (\Lambda_1 \Lambda_4 + \Lambda_2)$, and $\varrho_7 = \Lambda_4 \Lambda_6 + \Lambda_5$. To the authors' best knowledge, this is the first time that the Wei-Norman equations and explicit form of the TEO for this algebra have been calculated.

To obtain the CNOT gate in Eq. (100) from the above explicit TEO some constraints on the components of $\hat{U}$ must be satisfied. From $\hat{U}_{44}$, it is evident that obtaining the CNOT gate requires taking the limit $\Lambda_9 \rightarrow \infty$. To achieve the desired values for $\hat{U}_{34}$ and $\hat{U}_{43}$ in this limit, the conditions $\Lambda_6 = e^{\Lambda_9}$ and $\Lambda_{15} = e^{\Lambda_9}$ appear necessary. However, recalling that the unitarity constraints has not yet been applied to the TEO—and that the number of real variables currently exceeds the required 16—we introduce a phase ϕ_n through the definitions $\Lambda_6 = e^{\Lambda_9} e^{i\phi_n}$ and $\Lambda_{15} = e^{\Lambda_9} e^{i\phi_n}$, allowing for an additional flexibility and that will be determined later. To ensure the necessary zeros in the last row and column of the TEO as $\Lambda_9 \rightarrow \infty$, it suffices for the complex functions Λ_{13} , Λ_{14} , ϱ_6 , and ϱ_7 , to remain bounded. Additionally, achieving $e^{i\phi_n}$ as a global phase requires $\hat{U}_{11} = e^{i\phi_n}$ and $\hat{U}_{22} = e^{i\phi_n}$. Accordingly, we now derive the equations that arise from equating $\hat{U}$ and $\hat{U}_{\text{CNOT}}$ using the definitions $\Lambda_6 = e^{\Lambda_9} e^{i\phi_n}$ and $\Lambda_{15} = e^{\Lambda_9} e^{i\phi_n}$, and expressing the results in terms of ϱ_6 , and ϱ_7

$$\Lambda_{10} \Lambda_1 e^{(-\Lambda_7 + \Lambda_8)} + \Lambda_{11} (\Lambda_1 \Lambda_4 + \Lambda_2) e^{(-\Lambda_8 + \Lambda_9)} + \Lambda_{13} \varrho_6 e^{-\Lambda_9} + e^{\Lambda_7} = e^{i\phi_n}, \quad (\text{E19})$$

$$\Lambda_{12} (\Lambda_1 \Lambda_4 + \Lambda_2) e^{(-\Lambda_8 + \Lambda_9)} + \Lambda_{14} \varrho_6 e^{-\Lambda_9} + \Lambda_1 e^{(-\Lambda_7 + \Lambda_8)} = 0, \quad (\text{E20})$$

$$\varrho_6 e^{i\phi_n} + (\Lambda_1 \Lambda_4 + \Lambda_2) e^{(-\Lambda_8 + \Lambda_9)} = 0, \quad (\text{E21})$$

$$\varrho_6 = \Lambda_1 \Lambda_5 + \Lambda_3 + (e^{\Lambda_9} e^{i\phi_n}) (\Lambda_1 \Lambda_4 + \Lambda_2), \quad \text{and equivalently: } \varrho_6 = \Lambda_2 (e^{\Lambda_9} e^{i\phi_n}) + \Lambda_3 + \Lambda_1 \varrho_7, \quad (\text{E22})$$

$$\Lambda_{10} e^{(-\Lambda_7 + \Lambda_8)} + \Lambda_{11} \Lambda_4 e^{(-\Lambda_8 + \Lambda_9)} + \Lambda_{13} \varrho_7 e^{-\Lambda_9} = 0, \quad (\text{E23})$$

$$\Lambda_{12} \Lambda_4 e^{(-\Lambda_8 + \Lambda_9)} + \Lambda_{14} \varrho_7 e^{-\Lambda_9} + e^{(-\Lambda_7 + \Lambda_8)} = e^{i\phi_n}, \quad (\text{E24})$$

$$\varrho_7 e^{i\phi_n} + \Lambda_4 e^{(-\Lambda_8 + \Lambda_9)} = 0, \quad (\text{E25})$$

$$\varrho_7 = \Lambda_4 (e^{\Lambda_9} e^{i\phi_n}) + \Lambda_5, \quad (\text{E26})$$

$$\Lambda_{11} e^{(-\Lambda_8 + \Lambda_9)} + \Lambda_{13} e^{i\phi_n} = 0, \quad (\text{E27})$$

$$\Lambda_{12} e^{(-\Lambda_8 + \Lambda_9)} + \Lambda_{14} e^{i\phi_n} = 0, \quad (\text{E28})$$

$$e^{\Lambda_9} e^{2i\phi_n} + e^{(-\Lambda_8 + \Lambda_9)} = 0, \quad (\text{E29})$$

$$\Lambda_6 = e^{\Lambda_9} e^{i\phi_n}, \quad (\text{E30})$$

$$\Lambda_{13} e^{-\Lambda_9} = 0, \quad (\text{E31})$$

$$\Lambda_{14} e^{-\Lambda_9} = 0, \quad (\text{E32})$$

$$\Lambda_{15} = e^{\Lambda_9} e^{i\phi_n}, \quad (\text{E33})$$

where we used Eqs. (E22), (E26), (E30), and (E33), to explicitly express the definitions of ϱ_6 , ϱ_7 , Λ_6 , and Λ_{15} , respectively. The equation for $\hat{U}_{44}$ has been suppressed, as it becomes equivalent to taking the limit $\Lambda_9 \rightarrow \infty$ at the end of the calculations. From Eq. (E29) we have that $\Lambda_8 = -2i\phi_n + i\pi$. Accordingly, the equations involving this term, Eqs. (E19),

(E20), (E21), (E23), (E24), (E25), (E27), and (E28), are modified to

$$\begin{aligned} & -\Lambda_{10} \Lambda_1 e^{(-\Lambda_7 - 2i\phi_n)} - \Lambda_{11} (\Lambda_1 \Lambda_4 + \Lambda_2) e^{(2i\phi_n + \Lambda_9)} \\ & + \Lambda_{13} \varrho_6 e^{-\Lambda_9} + e^{\Lambda_7} = e^{i\phi_n}, \end{aligned} \quad (\text{E34})$$

$$\begin{aligned}
& -\Lambda_{12}(\Lambda_1\Lambda_4 + \Lambda_2)e^{(2i\phi_n + \Lambda_9)} + \Lambda_{14}\varrho_6e^{-\Lambda_9} \\
& - \Lambda_1e^{(-\Lambda_7 - 2i\phi_n)} = 0,
\end{aligned} \tag{E35}$$

$$\varrho_6e^{i\phi_n} - (\Lambda_1\Lambda_4 + \Lambda_2)e^{(2i\phi_n + \Lambda_9)} = 0, \tag{E36}$$

$$-\Lambda_{10}e^{(-\Lambda_7 - 2i\phi_n)} - \Lambda_{11}\Lambda_4e^{(2i\phi_n + \Lambda_9)} + \Lambda_{13}\varrho_7e^{-\Lambda_9} = 0, \tag{E37}$$

$$-\Lambda_{12}\Lambda_4e^{(2i\phi_n + \Lambda_9)} + \Lambda_{14}\varrho_7e^{-\Lambda_9} - e^{(-\Lambda_7 - 2i\phi_n)} = e^{i\phi_n}, \tag{E38}$$

$$\varrho_7e^{i\phi_n} - \Lambda_4e^{(2i\phi_n + \Lambda_9)} = 0, \tag{E39}$$

$$-\Lambda_{11}e^{(2i\phi_n + \Lambda_9)} + \Lambda_{13}e^{i\phi_n} = 0, \tag{E40}$$

$$-\Lambda_{12}e^{(2i\phi_n + \Lambda_9)} + \Lambda_{14}e^{i\phi_n} = 0, \tag{E41}$$

$$\Lambda_8 = -2i\phi_n + i\pi, \tag{E42}$$

where we wrote in Eq. (E42) the definition of Λ_8 . Multiplying Eqs. (E40) and (E41) by $e^{(-2i\phi_n + \Lambda_9)}$, and recalling that we are considering Λ_{13} and Λ_{14} as free parameters, we obtain

$$\Lambda_{11} = \Lambda_{13}e^{(-i\phi_n - \Lambda_9)}, \tag{E43}$$

$$\Lambda_{12} = \Lambda_{14}e^{(-i\phi_n - \Lambda_9)}. \tag{E44}$$

From Eqs. (E39) and (E26) it is clear that $\Lambda_5 = 0$. From Eqs. (E36) and (E22) we have that $\Lambda_1\Lambda_5 + \Lambda_3 = 0$, so

$\Lambda_3 = 0$. Using explicitly ϱ_6 and ϱ_7 , together with the above results, we can rewrite equations (E34)–(E39) as

$$-\Lambda_{10}\Lambda_1e^{(-\Lambda_7 - 2i\phi_n)} + e^{\Lambda_7} = e^{i\phi_n}, \tag{E45}$$

$$-\Lambda_1e^{(-\Lambda_7 - 2i\phi_n)} = 0, \tag{E46}$$

$$\varrho_6 = (e^{\Lambda_9}e^{i\phi_n})(\Lambda_1\Lambda_4 + \Lambda_2), \tag{E47}$$

$$-\Lambda_{10}e^{(-\Lambda_7 - 2i\phi_n)} = 0, \tag{E48}$$

$$-e^{(-\Lambda_7 - 2i\phi_n)} = e^{i\phi_n}, \tag{E49}$$

$$\varrho_7 = \Lambda_4(e^{\Lambda_9}e^{i\phi_n}), \tag{E50}$$

$$\Lambda_3 = 0, \tag{E51}$$

$$\Lambda_5 = 0, \tag{E52}$$

where we used the last two equations to explicitly write the previous findings. From Eqs. (E46) and (E48), we find that $\Lambda_1 = 0$ and $\Lambda_{10} = 0$. Substituting these results, Eq. (E45) yields $e^{\Lambda_7} = e^{i\phi_n}$, while Eq. (E49) gives $e^{\Lambda_7} = e^{-3i\phi_n + i(2n+1)\pi}$, where $n \in \mathbb{Z}$. By equating these expressions, we obtain $\phi_n = \frac{(2n+1)\pi}{4}$. Furthermore, to ensure $\hat{U}_{14} = 0$ and $\hat{U}_{24} = 0$ in the limit $\Lambda_9 \rightarrow \infty$, conditions $\Lambda_2 = \Lambda_4 = 0$ must be met. In summary, $\Lambda_6 = \Lambda_{15} = e^{\Lambda_9 + i\phi_n}$, $\Lambda_7 = i\phi_n$, $\Lambda_8 = i\pi(\frac{1}{2} - n)$, $\Lambda_9 \rightarrow \infty$, $\Lambda_{11} = \Lambda_{13}e^{(-i\phi_n - \Lambda_9)}$, $\Lambda_{12} = \Lambda_{14}e^{(-i\phi_n - \Lambda_9)}$, with $\phi_n = \frac{(2n+1)\pi}{4}$, while $\Lambda_1 = \Lambda_2 = \Lambda_3 = \Lambda_4 = \Lambda_5 = \Lambda_{10} = 0$. Λ_{13} and Λ_{14} remain as free bounded complex parameters.

-
- [1] J. J. Sakurai and J. Napolitano, *Modern Quantum Mechanics* (Pearson, Harlow, 2014).
 - [2] W. Magnus, On the exponential solution of differential equations for a linear operator, *Commun. Pure Appl. Math.* **7**, 649 (1954).
 - [3] S. Blanes, F. Casas, J. A. Oteo, and J. Ros, The magnus expansion and some of its applications, *Phys. Rep.* **470**, 151 (2009).
 - [4] J. Wei and E. Norman, Lie algebraic solution of linear differential equations, *J. Math. Phys.* **4**, 575 (1963).
 - [5] J. Wei and E. Norman, On global representations of the solutions of linear differential equations as a product of exponentials, *Proc. Am. Math. Soc.* **15**, 327 (1964).
 - [6] S. Charzyński and M. Kuś, Wei–Norman equations for a unitary evolution, *J. Phys. A: Math. Theor.* **46**, 265208 (2013).
 - [7] S. Charzyński and M. Kuś, Wei–Norman equations for classical groups, *J. Diff. Equ.* **259**, 1542 (2015).
 - [8] R. W. Brockett, System theory on group manifolds and coset spaces, *SIAM J. Control* **10**, 265 (1972).
 - [9] R. W. Brockett, *Lie Algebras and Lie Groups in Control Theory* (Springer, Netherlands, 1973), pp. 43–82.
 - [10] A. M. Bloch, Nonholonomic mechanics, in *Nonholonomic Mechanics and Control*, edited by S. S. Antman, L. Greengard, and P. Holmes (Springer, New York, 2015), pp. 235–313.
 - [11] R. M. Wilcox, Exponential operators and parameter differentiation in quantum physics, *J. Math. Phys.* **8**, 962 (1967).
 - [12] F. Dell’Anno, S. De Siena, and F. Illuminati, Multiphoton quantum optics and quantum state engineering, *Phys. Rep.* **428**, 53 (2006).
 - [13] J. F. Cariñena, A. Ibort, G. Marmo, and G. Morandi, *Geometry from Dynamics, Classical and Quantum* (Springer, Netherlands, 2015).
 - [14] R. Kosloff and Y. Rezek, The quantum harmonic Otto cycle, *Entropy* **19**, 136 (2017).
 - [15] S. Qvarfort, M. R. Vanner, P. F. Barker, and D. E. Bruschi, Master-equation treatment of nonlinear optomechanical systems with optical loss, *Phys. Rev. A* **104**, 013501 (2021).
 - [16] A. C. Duriez, D. Martínez-Tibaduiza, and A. Z. Khoury, Algebraic approach to a nonadiabatic coupled Otto cycle, *Phys. Rev. A* **106**, 062212 (2022).
 - [17] S. Qvarfort and I. Pikovski, Solving quantum dynamics with a Lie-algebra decoupling method, *PRX Quantum* **6**, 010201 (2025).
 - [18] L. Šnobl and P. Winternitz, *Classification and Identification of Lie Algebras* (American Mathematical Society, Providence, Rhode Island, USA, 2017), Vol. 33.
 - [19] R. Lecamwasam, T. Iakovleva, and J. Twamley, Quantum metrology with linear Lie algebra, *Phys. Rev. Res.* **6**, 043137 (2024).
 - [20] N. Hatano and M. Suzuki, *Finding Exponential Product Formulas of Higher Orders* (Springer, Berlin, 2005), pp. 37–68.
 - [21] L. Teuber and S. Scheel, Solving the quantum master equation of coupled harmonic oscillators with Lie-algebra methods, *Phys. Rev. A* **101**, 042124 (2020).
 - [22] D. Martínez-Tibaduiza, C. González-Arciniegas, C. Farina, A. Cavalcanti-Duriez, and A. Z. Khoury, Quantum-based

- solution of time-dependent complex Riccati equations, *Phys. Rev. A* **107**, 042211 (2023).
- [23] I. Ramos-Prieto, R. Román-Ancheyta, J. Récamier, and H. M. Moya-Cessa, Exact solution of a non-stationary cavity with one intermode interaction, *J. Opt. Soc. Am. B* **38**, 2873 (2021).
- [24] The original BCH formula can be straightforwardly deduced using the BCH lemma as $e^{\hat{g}_i} e^{\hat{g}_j} = \exp[\hat{g}_i + \int_0^1 d\theta e^{\theta \hat{g}_i} \hat{g}_j e^{-\theta \hat{g}_i}]$.
- [25] S. Barnett and P. M. Radmore, *Methods in Theoretical Quantum Optics* (Oxford University Press, 2002).
- [26] C. Altafini, Explicit Wei–Norman for matrix Lie groups via Putzer’s method, *Syst. Control Lett.* **54**, 1121 (2005).
- [27] R. Gilmore, Baker–Campbell–Hausdorff formulas, *J. Math. Phys.* **15**, 2090 (1974).
- [28] More generally, we have that $(\text{ad} \hat{g}_n) \cdots (\text{ad} \hat{g}_k)(\text{ad} \hat{g}_i) \hat{g}_j = \sum_{l=1}^L (\Upsilon^n \cdots \Upsilon^k \Upsilon^i)_{jl} \hat{g}_l$. This result could be useful to automate the construction of exact effective Hamiltonians [29].
- [29] A. Chakraborty, T. L. Patti, B. Khailany, A. N. Jordan, and A. Anandkumar, GPU-accelerated effective Hamiltonian calculator, *Quantum* **9**, 1946 (2025).
- [30] L. E. de Clercq, *et al.*, Estimation of a general time-dependent Hamiltonian for a single qubit, *Nat. Commun.* **7**, 11218 (2016).
- [31] K. Husimi, Miscellanea in elementary quantum mechanics, II, *Prog. Theor. Phys.* **9**, 381 (1953).
- [32] D. M. Tibaduiza, L. Pires, A. L. C. Rego, D. Szilard, C. Zarro, and C. Farina, Efficient algebraic solution for a time-dependent quantum harmonic oscillator, *Phys. Scr.* **95**, 105102 (2020).
- [33] D. Martínez-Tibaduiza, L. Pires, and C. Farina, Time-dependent quantum harmonic oscillator: A continuous route from adiabatic to sudden changes, *J. Phys. B: At. Mol. Opt. Phys.* **54**, 205401 (2021).
- [34] C. D. Ahlbrandt and A. C. Peterson, *Discrete Hamiltonian Systems* (Springer, 1996).
- [35] J.-S. Li and N. Khaneja, Ensemble control of Bloch equations, *IEEE Trans. Autom. Control* **54**, 528 (2009).
- [36] D. E. Bruschi and A. Xuereb, “Mechano-optics”: An optomechanical quantum simulator, *New J. Phys.* **20**, 065004 (2018).
- [37] D. E. Bruschi, Time evolution of coupled multimode and multiresonator optomechanical systems, *J. Math. Phys.* **60**, 062105 (2019).
- [38] S. Qvarfort, A. Serafini, A. Xuereb, D. Rätzel, and D. E. Bruschi, Enhanced continuous generation of non-Gaussianity through optomechanical modulation, *New J. Phys.* **21**, 055004 (2019).
- [39] C. Elouard, D. Herrera-Martí, M. Esposito, and A. Auffèves, Thermodynamics of optical Bloch equations, *New J. Phys.* **22**, 103039 (2020).
- [40] R. Dann, R. Kosloff, and P. Salamon, Quantum finite-time thermodynamics: Insight from a single qubit engine, *Entropy* **22**, 1255 (2020).
- [41] L. Dupays and A. Chenu, Shortcuts to squeezed thermal states, *Quantum*, **5**, 449 (2021).
- [42] O. Abah, G. De Chiara, M. Paternostro, and R. Puebla, Harnessing nonadiabatic excitations promoted by a quantum critical point: Quantum battery and spin squeezing, *Phys. Rev. Res.* **4**, L022017 (2022).
- [43] B. M. Mihalcea, Quasienergy operators and generalized squeezed states for systems of trapped ions, *Ann. Phys.* **442**, 168926 (2022).
- [44] Z. Fei and C. P. Sun, Many-body systems with SU(1, 1) dynamical symmetry: From dynamics to thermodynamics based on the trace formula, [arXiv:2111.11041](https://arxiv.org/abs/2111.11041).
- [45] C. C. Rusconi, M. Perdriat, G. Hétet, O. Romero-Isart, and B. A. Stickler, Spin-controlled quantum interference of levitated nanorotors, *Phys. Rev. Lett.* **129**, 093605 (2022).
- [46] D. F. Walls, Squeezed states of light, *Nature (London)* **306**, 141 (1983).
- [47] J. Ma, X. Wang, C. P. Sun, and F. Nori, Quantum spin squeezing, *Phys. Rep.* **509**, 89 (2011).
- [48] M. S. Silver, R. I. Joseph, and D. I. Hoult, Selective spin inversion in nuclear magnetic resonance and coherent optics through an exact solution of the Bloch–Riccati equation, *Phys. Rev. A* **31**, 2753 (1985).
- [49] R. A. de Graaf, *In Vivo NMR Spectroscopy: Principles and Techniques* (John Wiley & Sons, 2019).
- [50] J.-Q. Shen, H.-Y. Zhu, and P. Chen, Exact solutions and geometric phase factor of time-dependent three-generator quantum systems, *Eur. Phys. J. D* **23**, 305 (2003).
- [51] A. Laufer, The exponential map of GL(N), *J. Phys. A: Math. Gen.* **30**, 5455 (1997).
- [52] J. Bradbury, R. Frostig, P. Hawkins, M. J. Johnson, C. Leary, D. Maclaurin, G. Necula, A. Paszke, J. VanderPlas, S. Wanderman-Milne, and Q. Zhang, JAX: Composable transformations of Python NumPy programs, GitHub, 2018, <https://github.com/google/jax>.
- [53] A. Paszke, *et al.*, Pytorch: An imperative style, high-performance deep learning library (2019).
- [54] A. Meurer, *et al.*, Sympy: Symbolic computing in Python, *PeerJ Comput. Sci.* **3**, e103 (2017).
- [55] The name “block-decoupled” signals that some of the resulting differential equations for high-order QLSs will still be coupled, but in blocks with the maximum size of the algebra rank (maximum number of diagonalizable matrices), as we will see in Secs. III A and IV.
- [56] P.-Y. Tsai and C.-K. Chen, An approximate analytic solution of the nonlinear Riccati differential equation, *J. Franklin Inst.* **347**, 1850 (2010).
- [57] C. Altafini, Parameter differentiation and quantum state decomposition for time varying Schrödinger equations, *Rep. Math. Phys.* **52**, 381 (2003).
- [58] A. Das and S. Okubo, *Lie Groups and Lie Algebras for Physicists* (World Scientific, 2014).
- [59] D. H. Sattinger and O. L. Weaver, *Lie Groups and Algebras with Applications to Physics, Geometry, and Mechanics*, Applied Mathematical Sciences (Springer-Verlag, New York, 1986).
- [60] B. C. Hall, *Lie Groups, Lie Algebras, and Representations: An Elementary Introduction* (Springer International Publishing, Switzerland, 2015).
- [61] D. P. DiVincenzo, Two-bit gates are universal for quantum computation, *Phys. Rev. A* **51**, 1015 (1995).
- [62] A. Barenco, C. H. Bennett, R. Cleve, D. P. DiVincenzo, N. Margolus, P. Shor, T. Sleator, J. A. Smolin, and H. Weinfurter, Elementary gates for quantum computation, *Phys. Rev. A* **52**, 3457 (1995).
- [63] J. A. Bergou, M. Hillery, and M. Saffman, *Quantum Information Processing: Theory and Implementation* (Springer International Publishing, 2021).

- [64] D. A. Steck, Quantum and Atom Optics, available online at <http://steck.us/teaching> (revision 0.16.6, 17 October 2025).
- [65] <https://gitlab.com/VolodyaCO/dynamics-of-sun-systems>.
- [66] R. Loudon and P. L. Knight, Squeezed light, *J. Mod. Opt.* **34**, 709 (1987).
- [67] L. A. Wu, M. Xiao, and H. J. Kimble, Squeezed states of light from an optical parametric oscillator, *J. Opt. Soc. Am. B* **4**, 1465 (1987).
- [68] M. C. Teich and B. E. A. Saleh, Squeezed state of light, *Quantum Opt.* **1**, 153 (1989).
- [69] L. P. Grishchuk and Y. V. Sidorov, Squeezed quantum states of relic gravitons and primordial density fluctuations, *Phys. Rev. D* **42**, 3413 (1990).
- [70] C. F. Lo, Time evolution of scalar field in spatially flat Robertson-Walker space-time, *Phys. Scr.* **44**, 217 (1991).
- [71] J. Perina, *Quantum Statistics of Linear and Nonlinear Optical Phenomena* (Springer Science & Business Media, 1991).
- [72] L. P. Grishchuk, Quantum effects in cosmology, *Class. Quantum Grav.* **10**, 2449 (1993).
- [73] A. Albrecht, P. Ferreira, M. Joyce, and T. Prokopec, Inflation and squeezed quantum states, *Phys. Rev. D* **50**, 4807 (1994).
- [74] B. L. Hu, G. Kang, and A. Matacz, Squeezed vacua and the quantum statistics of cosmological particle creation, *Int. J. Mod. Phys. A* **09**, 991 (1994).
- [75] V. V. Dodonov, “Nonclassical” states in quantum optics: A “squeezed” review of the first 75 years, *J. Opt. B: Quantum Semiclass. Opt.* **4**, R1 (2002).
- [76] V. V. Dodonov and V. I. Man’ko, *Theory of Nonclassical States of Light* (CRC Press, 2003).
- [77] M. B. Einhorn and F. Larsen, Squeezed states in the de Sitter vacuum, *Phys. Rev. D* **68**, 064002 (2003).
- [78] V. V. Dodonov and A. V. Dodonov, Quantum harmonic oscillator and nonstationary Casimir effect, *J. Russ. Las. Res.* **26**, 445 (2005).
- [79] T. Fujii, S. Matsuo, N. Hatakenaka, S. Kurihara, and A. Zeilinger, Quantum circuit analog of the dynamical Casimir effect, *Phys. Rev. B* **84**, 174521 (2011).
- [80] P. Lähteenmäki, G. S. Paraoanu, J. Hassel, and P. J. Hakonen, Dynamical Casimir effect in a Josephson metamaterial, *Proc. Natl. Acad. Sci. USA* **110**, 4234 (2013).
- [81] S. Felicetti, M. Sanz, L. Lamata, G. Romero, G. Johansson, P. Delsing, and E. Solano, Dynamical Casimir effect entangles artificial atoms, *Phys. Rev. Lett.* **113**, 093602 (2014).
- [82] R. Schnabel, Squeezed states of light and their applications in laser interferometers, *Phys. Rep.* **684**, 1 (2017).
- [83] F. A. Raffa, M. Rasetti, and M. Genovese, New quantumness domains through generalized squeezed states, *J. Phys. A: Math. Theor.* **52**, 475301 (2019).
- [84] I. D. Leroux, M. H. Schleier-Smith, and V. Vuletić, Implementation of cavity squeezing of a collective atomic spin, *Phys. Rev. Lett.* **104**, 073602 (2010).
- [85] O. Hosten, N. J. Engelsen, R. Krishnakumar, and M. A. Kasevich, Measurement noise 100 times lower than the quantum-projection limit using entangled atoms, *Nature (London)* **529**, 505 (2016).
- [86] H. Bao, *et al.*, Spin squeezing of 10^{11} atoms by prediction and retrodiction measurements, *Nature (London)* **581**, 159 (2020).
- [87] H. Vahlbruch, S. Chelkowski, K. Danzmann, and R. Schnabel, Quantum engineering of squeezed states for quantum communication and metrology, *New J. Phys.* **9**, 371 (2007).
- [88] V. Giovannetti, S. Lloyd, and L. Maccone, Advances in quantum metrology, *Nat. Photon.* **5**, 222 (2011).
- [89] R. Slavík, *et al.*, All-optical phase and amplitude regenerator for next-generation telecommunications systems, *Nat. Photon.* **4**, 690 (2010).
- [90] K. G. Fedorov, *et al.*, Displacement of propagating squeezed microwave states, *Phys. Rev. Lett.* **117**, 020502 (2016).
- [91] S. Pogorzalek, *et al.*, Secure quantum remote state preparation of squeezed microwave states, *Nat. Commun.* **10**, 2604 (2019).
- [92] L. Barsotti, J. Harms, and R. Schnabel, Squeezed vacuum states of light for gravitational wave detectors, *Rep. Prog. Phys.* **82**, 016905 (2019).
- [93] M. Tse, *et al.*, Quantum-enhanced advanced LIGO detectors in the era of gravitational-wave astronomy, *Phys. Rev. Lett.* **123**, 231107 (2019).
- [94] W. Jia, V. Xu, K. Kuns, M. Nakano, and members of the LIGO Scientific Collaboration, Squeezing the quantum noise of a gravitational-wave detector below the standard quantum limit, *Science* **385**, 1318 (2024).
- [95] F. Wolf and H. J. Korsch, Time-evolution operators for (coupled) time-dependent oscillators and Lie algebraic structure theory, *Phys. Rev. A* **37**, 1934 (1988).
- [96] C. K. Law, Effective hamiltonian for the radiation in a cavity with a moving mirror and a time-varying dielectric medium, *Phys. Rev. A* **49**, 433 (1994).
- [97] Note that this type of calculation allows to obtain changes of basis or the time evolution of any Lie vector within the Heisenberg picture. More generally, for suitable well-behaved functions of the generators $f(\hat{g}_j)$, we have that $(e^{\Lambda_i \text{ad}_{\hat{g}_i}})f(\hat{g}_j) = f((e^{\Lambda_i \text{ad}_{\hat{g}_i}})\hat{g}_j)$ [25].
- [98] W. Pfeifer, *The Lie Algebras $Su(n)$* (Birkhäuser, Basel, 2003).
- [99] P. Skands, Introduction to QCD, in *Searching for New Physics at Small and Large Scales* (World Scientific, 2013).
- [100] H. Georgi and S. L. Glashow, Unity of all elementary-particle forces, *Phys. Rev. Lett.* **32**, 438 (1974).
- [101] M. A. Nielsen and I. L. Chuang, *Quantum Computation and Quantum Information: 10th Anniversary Edition* (Cambridge University Press, 2012).
- [102] R. Wiersema, D. Lewis, D. Wierichs, J. Carrasquilla, and N. Killoran, Here comes the $su(n)$: Multivariate quantum gates and gradients, *Quantum*, **8**, 1275 (2024).
- [103] C. M. Caves and G. J. Milburn, Qutrit entanglement, *Opt. Commun.* **179**, 439 (2000).
- [104] A. B. Klimov, R. Guzmán, J. C. Retamal, and C. Saavedra, Qutrit quantum computer with trapped ions, *Phys. Rev. A* **67**, 062313 (2003).
- [105] P. Gokhale, J. M. Baker, C. Duckering, N. C. Brown, K. R. Brown, and F. T. Chong, Asymptotic improvements to quantum circuits via qutrits, in *Proceedings of the 46th International Symposium on Computer Architecture, ISCA’19* (ACM Press, New York, NY, 2019), pp. 554–566.
- [106] Y. Wang, Z. Hu, B. C. Sanders, and S. Kais, Qudits and high-dimensional quantum computing, *Front. Phys.* **8**, 589504 (2020).
- [107] In case the inserted matrices do not close a Lie algebra, the user will be notified by the library.
- [108] G. Dattoli, P. Di Lazzaro, and A. Torre, $Su(1, 1)$, $su(2)$, and $su(3)$ coherence-preserving Hamiltonians and time-ordering techniques, *Phys. Rev. A* **35**, 1582 (1987).

- [109] M. F. Cavalcante, B. Çakmak, M. V. S. Bonança, and S. Deffner, Correcting noisy quantum gates with shortcuts to adiabaticity, *Europhys. Lett.* **152**, 48001 (2025).
- [110] Macbook Pro, Apple M1 Max chip, 32 GB RAM.
- [111] F. M. Surace, P. Fromholz, F. Scazza, and M. Dalmonte, Scalable ab initio protocol for quantum simulating $SU(N) \times U(1)$ lattice gauge theories, *Quantum* **8**, 1359 (2024).
- [112] R. Cestnik and E. A. Martens, Integrability of a globally coupled complex Riccati array: Quadratic integrate-and-fire neurons, phase oscillators, and all in between, *Phys. Rev. Lett.* **132**, 057201 (2024).
- [113] M. A. Lohe, Exact reduction of synchronized systems in higher-dimensional spaces, *Chaos* **35**, 023144 (2025).
- [114] A. Di Meglio, *et al.*, Quantum computing for high-energy physics: State of the art and challenges, *PRX Quantum* **5**, 037001 (2024).
- [115] D. M. Tibaduiza, L. Pires, A. L. C. Rego, D. Szilard, C. Zarro, and C. Farina, A time-dependent harmonic oscillator with two frequency jumps: An exact algebraic solution, *Braz. J. Phys.* **50**, 634 (2020).
- [116] D. Martínez-Tibaduiza, A. H. Aragão, C. Farina, and C. A. D. Zarro, New BCH-like relations of the $su(1, 1)$, $su(2)$ and $so(2, 1)$ Lie algebras, *Phys. Lett. A* **384**, 126937 (2020).
- [117] D. E. Bruschi, A. Xuereb, and R. Zeier, Deciding finiteness of bosonic dynamics with tunable interactions, *J. Phys. A: Math. Theor.* **58**, 025204 (2025).